

KOBRAHSOFT SOFTWARE

SP5 ADVANCED TAPE
TO +3 DRIVE
UTILITY

OWNERS MANUAL

Disc Version

(c). KOBRAHSOFT 1989

INDEX

<u>Page No.</u>	<u>Contents</u>
1	SP5 Utility; Introduction & Technical Section.
3	Spectrum Memory Map; Various memory areas.
4	Basic Program Area; "Peek Line".
5	Numbers; RANTOP; CLEAR; M/C CALLS; Microdrives.
6	Decimal and Hexadecimal Numbers.
10	Program Protection Methods; Tricks of the Trade.
15	The programs supplied - full details.
17	General Transfer Methods.
19	Examples of Transfers.
29	Speedlock Decoder SD1; Loading and Using.
32	Speedlock Decoder SD234; Loading and using.
33	Speedlock Decoder SD5; Loading and using.
34	Alkatraz Decoder AD1; Loading and Using.
36	Alkatraz Decoder AD2; Loading and Using.
37	Alkatraz Decoder AD3; Loading and Using.
38	Firebird Decoder FBI; Loading and Using.
39	DISKCAT Disc Catalogue program.
40	SP5 Tape to M/D Utility 128K.
46	KD1 Disassembler; Loading and Using.
47	Header Reader; Headerless Block Length Reader.

SP5 ADVANCED TAPE TO +3 DRIVE UTILITYINSTRUCTIONS FOR USE

NOTE :- THIS UTILITY IS SUPPLIED ON THE UNDERSTANDING THAT YOU USE IT TO TRANSFER YOUR OWN SOFTWARE TO +3 DRIVE - AND DO NOT USE IT TO MAKE COPIES TO DISTRIBUTE OR ILLEGALLY SELL. THIS IS PIRACY, AND WE DO NOT CONDONE PIRACY!

GENERAL INTRODUCTION.

SP5 is our latest utility package to help you transfer the MAJORITY of your software to +3 Drive. Unlike most similar packages, SP5 will help you transfer EVEN the latest PROTECTED programs. Its advanced features now include:-

- (1). ALL SP5 programs are easily loaded from disc via a quickly loaded Menu program.
- (2). Tackles even the latest protected machine code programs.
- (3). Now comes complete with our KD1 Disassembler.
- (4). Comes complete with our "SD1 SPEEDLOCK DECODER" program, which makes transfer of the "Old" type Pulsing Programs to +3 Disc much easier, combined with our new SD1Trans program (and AD1Trans - for AD1 - see below).
- (5). SP5 now contains our new SD234 program which means Speedlock Decoder for types 2, 3 and 4, and ALSO our Speedlock decoder SD5. These two decoders now enable the EASY transfer of even the latest Speedlock programs.
- (6). Also included now is our "AD1 ALKATRAZ DECODER" program, which allows the easy transfer of programs which load with an unusual screen and a counter to zero.
- (7). The latest additions to SP5 are our new AD2 and AD3 Alkatraz decoders which, like AD1, allow the transfer to +3 disc of the latest Alkatraz programs.
- (8). We have also included our NEW FBI program which transfers the multi-block Firebird programs to your +3 disc.
- (9). To complete a superb utility, SP5 now gives you our "DISKCAT" program, which gives a FULL catalogue of all the files on your discs, together with many other useful facilities such as Rename, selective Erase, Unerase and Examine, providing you with a very useful utility in its own right.
- (10). Now also contains instructions on how to transfer true 128K programs.

NOTE:- From here onwards, (ENTER) will mean "Press the ENTER Key".

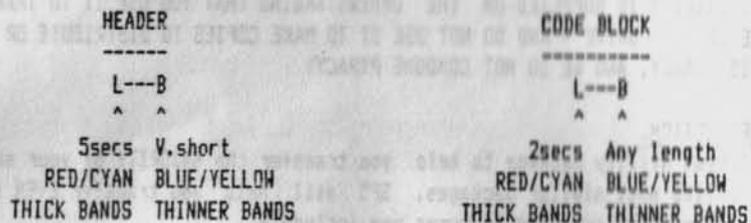
To get the most out of your SP5 utility - particularly if you are a relative novice to programming - we suggest you start with the Speedlock, Alkatraz and Firebird decoder programs, since these will allow you to transfer your Speedlock, Alkatraz and Firebird programs to your +3 Drive with the minimum of previous knowledge. For the more experienced programmer, the Technical Section below is a good place to start. In all cases, try transferring your OLDER games first, since they should be easier to transfer.

TECHNICAL SECTION.

When you load a program, you usually see at first a burst of RED/CYAN THICK STRIPES -

(2)

this is called a LEADER (L), and is around 5sec. in length for a "HEADER", but only 2sec. for a "CODE BLOCK". (These terms will be explained shortly). After, comes a burst of BLUE/YELLOW NARROWER STRIPES - these are BYTES (B) i.e. bits of code. Thus, for a typical BASIC program you get:-

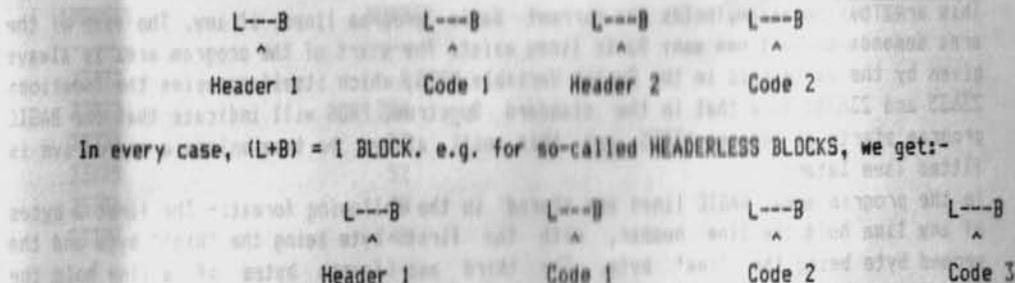


The HEADER must always come first, since it tells the Spectrum where the following code must go in memory. The burst of BYTES for a header is always very short, since it always contains only 17 bytes. These 17 bytes give the following information:-

BYTE NUMBER	INFORMATION
-----	-----
1	This gives the TYPE of program i.e. 0 for BASIC, 1 for a NUMERIC ARRAY, 2 for a STRING ARRAY, 3 for MACHINE CODE.
2-11	These 10 bytes give the PROGRAM NAME, in Spectrum character codes.
12,13	These 2 bytes give, for a block of code, the CODE LENGTH; or, for a Basic program, the length of the program area PLUS its variables.
14,15	These 2 bytes give, for a block of code, the Start Address of the block in memory; or, for a Basic program, the AUTO-RUN line number.
16,17	These 2 bytes give, for a block of code, a repeat of the CODE LENGTH; or, for a Basic program, the length of the program area MINUS its variables.

The information in these 17 bytes can be obtained using the Header Reader we have supplied. After the HEADER, usually comes a CODE BLOCK. This contains a leader of around 2sec. and then CODE which can be of any length. Thus, we can have:-

(3)



In every case, (L+B) = 1 BLOCK. e.g. for so-called HEADERLESS BLOCKS, we get:-

i.e. here we have 4 BLOCKS but only 1 HEADER - this can be identified by only having ONE leader of 5sec. - the rest will be 2sec. The length of Headerless Code Blocks can be obtained using our Headerless Block Length Reader (see later).

SPECTRUM MEMORY MAP.

In 48K Mode the +3 has a total memory capacity of 65535 Bytes or locations (i.e. 64K). This consists of 16K of ROM (Read Only Memory) - which contains the operating and Basic systems and cannot be written to or changed. Also, it has 48K of RAM (Random Access Memory) - which CAN be written to or changed. For a detailed diagram - see P.182 of your Plus 3 Manual. Consider the following main memory areas (48K Mode only):-

DISPLAY FILE AND ATTRIBUTES.

This area, (16384-23295), is the memory mapped area which forms the "high resolution" display of the Spectrum. Anything POKED into these locations will appear on your screen i.e. type POKE 16384,255 - and a small line will appear in the top L.H. corner of your screen. Note:- POKE - puts a value into a memory location, whereas PEEK - reads a value from a memory location. i.e. type POKE 16384,0 - the line disappears. Typing PRINT PEEK 0 gives the number 243 - the first byte in ROM.

PRINTER BUFFER.

The 256 locations from 23296 to 23551 stores any numbers destined for your printer. Without a printer, it is free. NOTE:- This is ONLY in 48K Mode - in 128K Mode this area is occupied by extra System Variables (see below).

THE EXTENDED SYSTEM VARIABLES.

In 48K Mode, the 182 locations from 23552 to 23733 contain the System Variables, whereas in 128K Mode an extended area of 437 bytes from 23296 to 23733 is used. Several of the variables are used in program protection (see later).

THE MICRODRIVE MAPS.

This is a dynamic area of RAM i.e. it varies in length e.g. when a microdrive operation is called, the area expands by up to 600 bytes to accomodate variables etc. (see later).

THE BASIC PROGRAM AREA.

This area of the memory holds the current Basic program lines, if any. The size of the area depends on just how many Basic lines exist. The start of the program area is always given by the value held in the System Variable PROG, which itself occupies the locations 23635 and 23636. Note that in the standard Spectrum, PROG will indicate that the BASIC program starts at address 23755, and this will always be true unless a microdrive is fitted (see later).

In the program area, BASIC lines are stored in the following format:- The first 2 bytes of any line hold the line number, with the first byte being the "high" byte and the second byte being the "low" byte. The third and fourth bytes of a line hold the "remaining length". This time the "low" byte comes before the "high" byte. The "remaining length" is the number of bytes from the fifth byte to the final ENTER character inclusively. Next, comes the BASIC line itself. Sinclair codes are used for the tokens and some characters. The last byte of a line is always an ENTER character (13). If a decimal number occurs in a BASIC line, it is stored as its ASCII characters and followed by the NUMBER character (14), plus 5 more bytes containing the Floating-Point form of the number. Type in the following:-

```
10: FOR A=23755 TO 24000:PRINT A,:PRINT PEEK A:NEXT A
```

Now RUN it - it will print the addresses and characters contained in its own line i.e.:-

ADDRESS	NUMBER	TOKEN / CHARACTER
23755	0) Line No. = (256+0)
23756	10) +10 = 10.
23757	38) Line Length=(256 +
23758	0) 0)+38 = 38.
23759	235	FOR
23760	65	A
23761	61	=
23762	50	2
23763	51	3
23764	55	7
23765	53	5
23766	53	5
23767	14) 6 byte F.P. No.
23768	0) = (92*256)+ 203
23769	0) = 23755
23770	203)
23771	92)

ADDRESS	NUMBER	TOKEN / CHARACTER
23772	0)
23773	204	TO
23774	50	2
23775	52	4
23776	48	0
23777	48	0
)))
()	(
23796	13	ENTER

The above program (PEEK-LINE) is useful for examining a range of memory locations.

NUMBERS.

We have talked above about 2 byte numbers and "high" and "low" bytes. The biggest number your Spectrum can contain in a single byte (location) is 255. So how can it handle bigger numbers? The answer is that it splits the number into 2 parts (for numbers from 0 to 65535). To calculate the number, you add the FIRST (or low order byte - L.O.B) to 256 x the SECOND (or high order byte - H.O.B). i.e. above:- LINE NUMBER = 10 + (256*0) = 10. Also, for the Floating Point numbers:- NUMBER = 203 + (92*256) = 23755. The L.O.B and H.O.B are usually stated, so you should not get them the "wrong way round".

RAMTOP AND CLEAR.

The main purpose in typing, say, CLEAR 29999, is that all the memory locations from 30000 up to the top of memory (65535), will be reserved - say for some machine code. The number here, 30000, is called RAMTOP - it is always 1 above the current CLEAR statement - it is the upper limit for your Basic program and its variables. NOTE:- Typing NEW will clear the memory up to RAMTOP but NOT above - this area will now only be cleared by:- (1) Disconnecting the mains supply at the back of your Spectrum, or (2) A better way (and less wearing on the Spectrum) is to type:- RANDOMISE USR 0. This has the same effect - it is said to RESET the computer. Any machine code is best placed above RAMTOP, since it is then out of the dynamic Basic area, which can otherwise be moved around in memory by various peripherals e.g. microdrive etc.

MACHINE CODE CALLS.

Machine code routines are usually executed using RANDOMISE USR number (R.USR n); or perhaps; PRINT USR number e.g. R.USR 0 above. This RESETS the computer by calling the machine code routines in ROM which start at address 0.

MICRODRIVES.

The main point to remember, is that when a microdrive is fitted, it can move up the

start of the Basic program area (usually 23755). This is because the drive needs 58 more system variables than normal i.e. the value of PROG will be $23755 + 58 = 23813$. However, since these extra variables are only invoked if a microdrive is fitted, this problem does not affect us when using the +3 Drive i.e. the value of PROG remains the same at 23755, irrespective of whether the drive has been used or not. This in itself helps make transferring programs to +3 Drive easier than transferring them to Microdrive, since this problem does not occur.

NOTE:- Here, again, PROG is a 2 byte number, so we have:- $PROG = \text{Value at } 23635 + 256 \times \text{Value at } 23636$. Where 23635 is the L.O.B; 23636 is the H.O.B.

DECIMAL AND HEXADECIMAL NUMBERS.

When we count in the usual DECIMAL way we say:-

0,1,2,3,4,5,6,7,8,9

Then:- $9+1 = 10$ or:-

10x10x10 (Thousands)	10x10 (Hundreds)	10x1 (Tens)	10x0 (Units)
0	0	0	9
			+1
			1 0

i.e. the Decimal system is based on blocks of TEN. Similarly, the HEXADECIMAL system is based on blocks of SIXTEEN. Here:-

HEXADECIMAL:- 0,1,2,3,4,5,6,7,8,9,A, B, C, D, E, F

DECIMAL :- 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15

Then:-

16x16x16 (4096)	16x16 (256)	16x1 (16)	16x0 (1)
0	0	0	F (15 Dec.)
			+ 1
			1 0 16

As you can see, the main difference are the letters A to F representing the numbers 10 to 15. Consider the decimal number FIVE THOUSAND ONE HUNDRED AND TWO - how is this represented numerically? First, we divide the number by the largest possible of the

numeric blocks - here 1000. This gives $5 \times 1000 = 5000$, remainder = 102. Next, divide by 100 - this gives $1 \times 100 = 100$, remainder = 2 i.e 0 tens and 2 units. Thus:- $5102 = 5 \times 1000 + 1 \times 100 + 0 \times 10 + 2 \times 1$. Now, what is this in Hexadecimal? Here, the blocks are:-

$16 \times 16 \times 16 \times 16 = 65536$ $16 \times 16 \times 16 = 4096$ $16 \times 16 = 256$ $16 \times 1 = 16$ $16 \times 0 = 0 - F$

Here:- $5102 / 4096 = 1$ remainder = 1006

$1006 / 256 = 3$ remainder = 238

$238 / 16 = 14 (= E)$ remainder = 14 (=E).

Thus, 5102 D = 13EE H. D and H represent Decimal and Hexadecimal.

Why do we need Hex. (for short) numbers, you may well ask? The reason is that most disassemblers (our KD1 too!) use Hex notation in their disassemblies. Also, Hex numbers are often used in Machine Code routines. We suggest you practice converting Hex to Decimal numbers and vice-versa until you feel you can understand them thoroughly. If in doubt as to what is a correct answer - use the number converter in our KD1 disassembler supplied.

THE MACHINE STACK.

This is the area of the computer's memory which is used to manipulate numbers. It is MOST IMPORTANT to always be aware of the location of the stack in memory, since, if it should get overwritten, the computer will almost certainly "lock-up" or crash. (See RAMTOP and CLEAR). A CLEAR instruction e.g. CLEAR 29999, ensures here that the stack is set below RAMTOP - thus any machine code above this address will NOT overwrite the stack. (See also machine code instructions, later).

INTRODUCTION TO MACHINE CODE.

Don't worry! It's not as difficult as you may think! In order to transfer the latest programs, a rudimentary knowledge of machine code is really required, since they are protected by various machine code routines.

WHAT IS MACHINE CODE?

Machine code is simply a sequence of numbers which the Z80 processor in your Spectrum can "understand", i.e. it translates the numbers into instructions which it then performs. The main reasons why machine code is hard to use are that (1) Any error in the number sequence will send a completely different instruction from the one you intended - usually crashing the computer. (2) There are no obvious error warnings. (3) There is no simple relation between the numbers and English i.e. as in Basic. Having frightened you thoroughly, let us say that we intend explaining just a few of the MAIN instructions - enough for you to tackle transfers to your +3 Drive!

YOUR PROGRAMS AND WHAT THEY DO.

For your assistance, we have supplied a large suite of programs. Mainly, they are in Basic - these will POKE some machine code into a particular location, this is then executed. A few are pure machine code - mainly Header and Block length readers - you need not understand how these work! NOTE:- These programs are ALL our copyright, so please don't sell them or give them to friends. We have deliberately left the programs visible, so that you can, if you wish, examine them. However, we HAVE included other copyright protections which, of course, we cannot disclose!

A FEW SIMPLE MACHINE CODE INSTRUCTIONS.

The Z80 processor has THREE main sets of storage locations called REGISTERS. These are the HL, DE and BC registers. Each register can store a number from 0 to 65535. The letters chosen are arbitrary - but HL will help you remember how the number is stored. Remember, a number from 0 to 255 can be stored in ONE BYTE (location). A number from 256 to 65535 needs TWO BYTES - a H.O.B. and a L.O.B. In the HL register:- the H register stores the H.O.B; the L register stores the L.O.B. Similarly for DE and BC. Each register pair, HL, DE or BC can be used singly i.e. D and E, H and L, or B and C. Another register pair also exists - AF. Here, A is called the ACCUMULATOR. F is called the FLAGS register. There is also an INDEX REGISTER - IX. This is used as a double register only i.e. you can't have I and X separately. The machine code instructions consist of numbers which are loaded into these registers.

NUMBER STORAGE IN THE Z80.

Another complication is that in machine code, numbers are stored in a "REVERSE" order! i.e. not as H.O.B first, L.O.B second, but L.O.B FIRST; H.O.B SECOND! This is just a convention, but it is important to remember it. Thus, suppose we have a number whose H.O.B = 82, and L.O.B = 100. As we showed you earlier, the actual number is:- $100 + 256 * 82 = 21092$. This, however, is stored in machine code "IN REVERSE" i.e.:- ----100, (L.O.B)82, (H.O.B)----.

INSTRUCTIONS.

Below are the main instructions you will need, together with the machine code numbers which represent them. NOTE:- A SINGLE BYTE number = n, a TWO BYTE number = nn, or nH,nL (H.O.B, L.O.B).

MNEMONIC	INSTRUCTION	NUMBERS.
LD HL,nn	LOAD HL register with 2 byte number	33,nL,nH
LD DE,nn	LOAD DE register with 2 byte number	17,nL,nH
LD BC,nn	LOAD BC register with 2 byte number	1,nL,nH
LD IX,nn	LOAD IX register with 2 byte number	221,33,nL,nH
LD A,n	LOAD ACCUMULATOR with 1 byte number	62,n

MNEMONIC	INSTRUCTION	NUMBERS
SCF	SET CARRY FLAG	55
CALL nn	CALL routine at nn	205,nL,nH
RET	RETURN from routine	201
JP nn	JUMP to address nn	195,nL,nH
RST 0	RESTART ZERO	199
ADD HL,DE	ADD contents of DE to contents of HL	25
CP n	COMPARE number n with number in A	254,n
DEC HL	DECREMENT contents of HL register by 1	43
DEC BC	DECREMENT contents of BC register by 1	11
DEC DE	DECREMENT contents of DE register by 1	27
INC HL	INCREMENT contents of HL register by 1	35
INC BC	INCREMENT contents of BC register by 1	3
INC DE	INCREMENT contents of DE register by 1	19
LD (DE),A	LOAD contents of A into address pointed to by DE	18
LD A,(DE)	LOAD contents of address pointed to by DE into A	26
LDIR	LOAD, INCREMENT AND REPEAT	237,176
SBC HL,DE	SUBTRACT contents of DE from contents of HL	237,82

These are the main instructions. In total, for the Z80, there are over 600! NOTE:- MNEMONIC is the "shorthand" version of the instruction as shown in books, assemblers etc. (Pronounce it "NEEMONIC"). The LDIR instruction is special - it is used to move blocks of memory around i.e.:- HL contains the "start" address; DE contains the "destination" address; BC contains the "number of bytes to move". What it does is - load the contents of HL into address DE, then decrement BC - if this is not zero, increment HL and DE and repeat - hence "LOAD INCREMENT AND REPEAT". So whenever you see an LDIR instruction (or the numbers 237,176) - you know a block of memory is being moved. This is used as a method of protection in some programs - see later.

MACHINE CODE "LOAD" TECHNIQUES.

To help with +3 Drive transfer, you must learn to recognise the sequence of instructions (numbers) frequently used in programs to LOAD in blocks of code. These are:- (1) SCF. (2) LD A,n. (3) LD IX,nn. (4) LD DE,nn. (5) CALL nn. (6) RET. Thus, to load a block of code using machine code, the sequence is:- (1) Set the carry flag - this signals "LOAD" (55). (2) Load the accumulator with a number - this is either 0 (to load a Header), or 255 (to load a code block). i.e. we can have:- 62,0 OR 62,255. (3) Load IX with a number - this indicates the START ADDRESS (S.A.) of the block in memory i.e. 221,33,nL,nH. (4) Load DE with a number - this is the CODE LENGTH (C.L.) of the block i.e. 17,nL,nH. (5) Call address nn. This is USUALLY (but not always) the address of the LOAD routine in ROM, and = 1366 i.e.:- 205,86,5. (address = $86 + 5 * 256 = 1366$). (6)

Return from this routine i.e. 201. So, a typical sequence to LOAD would be:- 55,62,n,221,33,nL,nH,17,nL,nH,205,06,5,201, or any combination of these numbers. A typical "load game" structure is:- (1) BASIC LOADER. (2) LOAD SCREEN* (picture). (3) LOAD GAME CODE. (4) JUMP TO START ADDRESS. In machine code, this would appear as:-

```

SCF      - signal "LOAD".
LD A,0   - signal load HEADER.
LD IX,nn - START address
LD DE,17 - LENGTH = 17 bytes for a HEADER.
CALL 1366 - LOAD it.
SCF      - signal "LOAD".
LD A,255 - signal load "CODE".
LD IX,16384 - start address = start of screen RAM.
LD DE,6912 - length = length of screen RAM.
CALL 1366 - LOAD it.
SCF      - signal "LOAD".
LD A,255 - signal load "CODE".
LD IX,nn - START address of code.
LD DE,nn - LENGTH of code.
CALL 1366 - LOAD it.
JP nn    - JUMP to start address.

```

Now we know "what to look for", let's use this in conjunction with the methods used to protect modern programs. NOTE:- Use 48K Mode except where 128K Mode is specified.

PROGRAM PROTECTION METHODS.

Hints and Tips, Tricks of the Trade.

(1). Auto-Run in Basic.

Early Basic programs were protected by making them AUTO-RUN when reloaded with the LOAD "" command. This was done by saving them with:- SAVE"name" LINE n. Where n was the line number to start from. However, this is simply stopped by using:- MERGE "" instead of LOAD "". When the Basic loads, the auto-run is stopped and it can be listed as usual. NOTE:- You can similarly save a program to +3 Drive with the command SAVE"name"LINE n. This will auto-run on reloading with LOAD "". However, it CAN be stopped by using MERGE "" with the +3 Drive, it CANNOT with a microdrive.

Invisible Lines, etc.

A better protection for Basic programs is to insert the line:- POKE 23659,0 - near the start of your program. Any attempt to BREAK into the program will cause a crash - try it! Type:-

```

10: REM protected program
20: POKE 23659,0
30: PRINT "PROTECTED"
40: GOTO 40

```

Now, save to auto-run with:- SAVE "test" LINE 10. Next, load back, using LOAD "". The word PROTECTED appears. Now press BREAK - the computer crashes! Reset - by pressing the RESET button on the left hand side. The reason is that location 23659 is a System Variable called DFSZ - it tells the Spectrum the number of lines available for printing messages at the bottom of the screen. This is usually = 2. Try:- PRINT PEEK 23659 - you should get 2. If you change this to zero, no lines are available and the computer crashes. However, using MERGE, the program can still be stopped and listed! NOTE:- Line 40 stops message print out and stops an immediate crash. So now we will make our program lines INVISIBLE! Type:-

```

5: POKE 23659,0
10: PRINT "This"
20: PRINT "is"
30: PRINT "Invisible"
40: GOTO 40

```

Enter EDIT mode (Caps Shift/I); edit each line. While in EDIT mode, press CAPS SHIFT and SYMBOL SHIFT to enter extended mode. Then, PRESS (CAPS SHIFT and 7),ENTER - the line disappears! Repeat with each line. A LIST now shows just the number 5 - the first line number! To see how this works, type:- PRINT PEEK 23635+256*PEEK 23636. This gives the start of Basic as 23755 as we explained earlier. Now type:-

```
50: FOR A=23755 TO 65535:PRINT A,:PRINT PEEK A:NEXT A
```

Now type:- RUN 50. This shows the above Basic lines. NOTE:- at 23759,23760 and 23786,23787 we see 16,7 i.e. INK 7. This is the clever part, since the result is that it makes PAPER and INK = 7 and makes the line invisible. Another method is to put a copyright line in your program (usually line 1) then POKE it to zero. Thus, type:- 1: REM This is MY Copyright. Now, find PROG as above. The first line number will occur at 23755+1=23756. Thus, type:- POKE 23756,0. Next, we LIST the program. Your copyright is now in line 0 - and it can't be edited out - try it! This program could now be saved to auto-run. It can still be stopped and listed using MERGE, but the lines will be invisible! NOTE:- By checking PROG and listing the program lines - you could also POKE all the other line numbers to zero! NOTE:- Keep this program for (3) below.

(7). Other Protection Methods.(a). Altering System Variables.

Some programs load different values than normal into the System Variables. Popular locations are DFSZ (see earlier), and also ERR-SP. This is the return address when an error occurs. It usually contains the numbers 84 (L.O.B) and 255 (H.O.B). These are stored at 23613, 23614 - the address of ERR-SP. This gives the address 65364 (i.e. $84 + 256 \times 255$) and this gives the error return address as 4867 - the normal return to Basic. Check with:- PRINT PEEK 23613 + 256*PEEK 23614 - this should give 65364. Also, PRINT PEEK 65364 + 256*PEEK 65365 - this should give 4867. (i.e. $3 + 256 \times 19 = 4867$). The usual ruse is to POKE ERR-SP with the address 23728 - where 23728, 23729 are 2 unused System Variable locations, normally containing 0. Thus, POKE 23728, 176; POKE 23729, 92 is usually used i.e. the address is $176 + 256 \times 92 = 23728$. Since these locations are 0, any error which would normally print an error message i.e. D-BREAK CONT REPEATS, Out of Memory, etc. would direct the Z80 to address 0 i.e. the computer would reset (as with R.USR 0 - remember?). Another System Variable used is FRAMES. This is 3 locations i.e. 23672, 23673, 23674. It acts as a sort of "clock" - its value is incremented every 20 milliseconds by the Z80. Some programs load values into FRAMES, then recheck that they are within certain limits when the program has loaded. If not it usually resets the computer. Many of the latest programs overwrite the System Variables - take care if this is the case - since trying to return to Basic will usually crash the computer.

(b). Checking the Screen.

Several programs check the SCREEN\$ picture loaded, and if it has been altered, reset the computer. These are in the minority however, and in most examples the picture can be left out to save disc space.

(c). Moving Code Around.

Many programs, after loading, move parts of the code to different areas of memory - using the LDIR command (see earlier).

(d). Scramblers.

Some programs, notably the U.S Gold series, employ SCRAMBLERS i.e. the main code, when loaded in, if checked would seem to be rubbish; but there is usually a small block of code somewhere which rearranges or UNSCRAMBLES the whole code - transforming it back to normal code. The problem usually, is finding the unscrambler! However, many are near the end of the main code.

OBJECTS AND PROBLEMS IN CONVERSIONS.

The programs encountered fall into 3 main types:-

(1). BASIC PROGRAMS.

If unprotected - these are simply transferred to +3 Drive using the command SAVE"name", OR:- Also use SAVE "NAME" Line Number to auto-run. If protected, e.g. auto-running: stop with MERGE (see earlier), and transfer as above.

(2). UNPROTECTED MACHINE CODE PROGRAMS.

To test for protection, proceed as follows:- Firstly, try loading the Basic Loader (the FIRST part) using MERGE **. If unprotected, it will stop any auto-run, allowing the program to be LISTED as normal. If this causes a crash, the program is quite probably protected. Also, if the game code can be loaded and run separately, using the Execution Address (from the Basic Loader), it is probably unprotected. With these programs, the code can usually be easily relocated if required, and transferred to +3 Drive.

(3). PROTECTED MACHINE CODE PROGRAMS.

These are in the majority nowadays and are of course the most difficult - needing a small knowledge of machine code to transfer them. Here, the Basic Loader(s) MUST be loaded checked and listed with MCBasldr and BASCONV. Remember, EVERY program, protected or not, MUST start with a normal speed Basic Loader.

With types (2) and (3), the main object is to get the code into memory (i.e. it must be prevented from auto-running), where it can be relocated as required, then transferred to +3 Drive.

PLEASE NOTE:- All transfer operations may be done in +3 Basic, EXCEPT those which may involve putting any code in the Printer Buffer area (see earlier). Any such transfers MUST be done in 48K Mode - if attempted in +3 Basic a "crash" will almost invariably occur, since in +3 Basic this area is occupied by extra System Variables which are used by the computer. This is not the case in 48K Mode. Also, it is wise to try and load a program in +3 Basic BEFORE attempting to transfer it to disc, to ensure it is compatible with +3 Basic - some of the older programs are NOT compatible with the +3!

THE PROGRAMS SUPPLIED AND WHAT THEY DO.

We have supplied you with a large suite of programs on your SP5 disc to help you with your transfers. These are listed below.

LOADING INSTRUCTIONS FOR PROGRAMS.

To load a program from your SP5 disc, proceed as follows. RESET the +3 by pressing the reset button on the left side of the computer. Place the SP5 disc in the drive and, at the Main Menu, press the ENTER key. A special Menu type program will load displaying the program names and numbers. The message "Enter your choice:-" also appears. To load a specific program, it is then only necessary to press the number key of the program required i.e. 1 to 23, followed by pressing ENTER. The selected program will load and run automatically. The programs available are:-

HEADER READER.

Probably the most useful program. To load it, type 8 then ENTER. This tells you the composition of the program you wish to transfer. (see later for fuller details). It gives the Program Name, Program Type, Start Address, Length, etc.

HEADERLESS BLOCK LENGTH READER.

To load, type 7 then (ENTER). This will tell you the length of any program without a Header i.e. the length of any headerless block e.g. for a Header, it will find 17 bytes. (useful for detecting false Headers, which are often not 17 bytes).

MCbasldr.

To load, type 11 then (ENTER). It is a "Machine Coded Basic Loader" i.e. it loads (via machine code) those "Basic Loaders" which contain machine code and crash when loaded via MERGE **. The program puts the code (Basic and machine code) at address 32000, where it can then be examined with your next program:-

BASCONV.

To load, press key 3 then ENTER. It is a BASIC CONVERTER, it reads code from (3) above - "lists" any Basic and gives the address of the 1st byte after any REMS (which may contain machine code). It is extremely versatile in that it will also make visible any Basic lines which have been rendered invisible (see earlier); it also gives the TRUE value for any numeric values found, since these may have been altered to give false values. The TRUE numeric value is the one in brackets. NOTE:- Pressing "p" gives a listing to a printer but in +3 mode (i.e. 128K) only. Any machine code in the REMS may be examined with:-

KD1 DISASSEMBLER.

To load, press key 9 then ENTER. It is an important addition to your SP4, since you now have the facility to load in machine code, and get a full disassembly of all the instructions - see later for full instructions.

CODE-STOP.

To load, press key 5 then ENTER. It is a machine code program which, when added to the end of an auto-running code block, stops it from running and returns to Basic, leaving the code in memory.

MOVEUP, MOVEDOWN, BLOCK MOVE.

To load, press keys 13 or 12 or 4 then ENTER. As suggested - these programs move blocks of code around in memory.

SPEEDLOCK DECODER SD1, SD1 Trans.

To load press key 14 or 15 then ENTER. SD1 is a useful Speedlock decoder program which enables the EASY transfer to +3 Drive (and a backup to tape, if required) of the "Pulsed Leader" type of program. SD1 Trans is a new program which enables the easy transfer to disc of the tape produced from SD1. Now with SD1, even the latest Pulsing programs e.g. "LEADERBOARD", "ARKANOID", "ENDURO RACER" are easily transferred to disc.

SPEEDLOCK DECODER SD234; SLOCKLDR.

To load, press key 6 or 16 then ENTER. SD234 stands for "Speedlock Decoder for types 2,3 and 4"; SLOCKLDR stands for "Speedlock Loader" - it is an automatic loader for SD234 transferred programs.

SPEEDLOCK DECODER SD5; SD5LDR.

To load, press 17 or 18 (ENTER). This decodes the latest type 5 Speedlock programs using a "software emulator" for the decoding. SD5LDR is a loader for SD5 transferred programs.

ALKATRAZ DECODER AD1; AD1 TRANS; LOADER.

To load, press key 1 or 2 or 10 then ENTER. With AD1 you can transfer programs protected by the Alkatraz system. AD1 TRANS effects simple transfer of the tape obtained from AD1; LOADER is an auto loader program for same.

ALKATRAZ DECODERS AD2; AD3; TRANSFER.

To load press keys 19, 22 or 20 then ENTER. AD2 and AD3 are our NEW Alkatraz decoders, similar to AD1. TRANSFER transfers the tapes from these AND FBI (see below) to disc.

FIREBIRD DECODER FBI.

To load press keys 23 then ENTER. This NEW program will transfer the multi-block Firebird programs which load with many very small blocks, to disc.

DISKCAT - A FULL DISC CATALOGUE PROGRAM.

To load press keys 21 then ENTER. With this superb program you can now get a FULL catalogue VERY QUICKLY of ALL the files on your discs.

GENERAL METHODS OF TRANSFER.

The main point to remember is that in ANY game, the first part to load (usually a Basic program), MUST be normal speed. The 4 main program types we will encounter are:-

(1). UNPROTECTED BASIC PROGRAMS.

Load from tape with LOAD ** then PLAY, then save to +3 Drive with:- SAVE"name" OR SAVE"name" Line no. to auto run.

(2). PROTECTED BASIC PROGRAMS.

Stop any auto run by loading using MERGE** - save as in (1) above.

(3). UNPROTECTED MACHINE CODE PROGRAMS.

These are in the minority nowadays. The only programs which we know of which have unprotected machine code today are those by Melbourne House, and some budget games.

(4). PROTECTED MACHINE CODE PROGRAMS.

This covers the vast majority of modern games. Unfortunately, they can also be very difficult to transfer to +3 Drive, mainly because of the many and varied protection methods used. We hope you will appreciate that any utility like SP5 CANNOT transfer ALL programs; but it can give general methods to transfer MOST.

We will now consider the main types (3) and (4) in more detail:-

TRANSFER OF UNPROTECTED MACHINE CODE PROGRAMS.

A general (but necessarily simplified) method of transfer for BOTH unprotected AND protected programs is as follows:-

(1). Load the Basic Loader, using MERGE ** for unprotected programs. For protected programs, we advise you to load it thus:- First, load in the "Header Reader" program, then PLAY in the Basic Loader. Note the total length. This gives the number of bytes to load in using the MCBasldr program. Reset the Spectrum by pressing RESET - this will remove the Header Reader program and clear the computer. Next, load the MCBasldr program. Type in the figure you have noted for "Number of bytes to Load = ". Follow the on screen instructions, then PLAY in the Basic Loader. The MCBasldr program will load ANY Basic Loader to address 32000, where it can then be examined using the BASCONV program. As instructed on screen, press any key to NEW the Spectrum - this removes the MCBasldr program but the Basic is still intact at address 32000. Press ENTER then key 3 to load BASCONV; this program will "LIST" the Basic Loader, making any invisible lines visible. It also gives the address of any REM statements found, since these can be used to hold machine code (see earlier). Also, it gives the TRUE values for any numeric values found - the true value is the one in brackets. NOTE:- The initial screen asks if you require output to the screen (press "S"), or to a printer (press "P"). To output to a printer, since the +3 is in 128K mode, simply press "P", and the listing will be printed. It then asks you the Start Address of the Code - for a normal Basic Loader this is usually 23755, but it can be 23813 if a microdrive was originally used.

Next, enter the number of bytes to analyse - this is the value you noted using the Header Reader program. The Basic is then "listed" for you. Note any important items such as CLEAR statements, or POKEs or RANDOMISE USR values; also any PAPER or INK or BORDER colours. If you see any REM statements followed by a series of incomprehensible codes e.g. STEP ^, / FOR INVERSE ! etc, this is probably machine code - note the address of the first character after the REM (given in the program) - this will be the start of the machine code which could possibly load the game.

To investigate any such code, type CLEAR 30000, then NEW. This will clear the BASCONV program, leaving the code intact. Press ENTER then key 9 to load KSFT KD1, and disassemble from the start of the code. Look for any hints as to the start and length and methods of loading the game code blocks (see earlier).

We have covered this part in much detail, since it will be frequently used. Next:-

- (2). Look for the method of protection used in part (1) above.
- (3). Decode the protection, and isolate the code to load the game.
- (4). Stop the code from running after it has loaded the game so it returns to Basic.
- (5). Load the game code, ensuring a return to Basic so it can be saved to +3 DRIVE.
- (6). Move and/or save the game code to +3 Drive as required.
- (7). Save a Basic Loader to reload the game code and start the game.

NOTE:- For unprotected programs, steps (2), (3) and (4) will not be needed. The whole object is to isolate the code which loads the game, make it return to Basic when it has loaded the game; save the game code to tape, then transfer it to +3 Drive, together with a Basic Loader. As an example to illustrate the method, consider:-

WAY OF THE EXPLODING FIST. (48K).

Press RESET and load the SPS Menu. Press 0 (ENTER) to load the HReader program. PLAY in the "FIST" tape from the start until all the headers have been read. This gave:-

Tape Count	File Name	Type	Details
6 - 9	FIST	BASIC	T.L.=319,P.L.=319,A.R. Line 10
10 - 23	t	M/C	S.A.=32768, C.L.=6912 (SCREEN#)
24 - 107	k	M/C	S.A.=24576, C.L.=38912

Where:- T.L.= Total Length, P.L.= Program Length, S.A.= Start Address, C.L.= Code Length, A.R.= Auto-Run Line No., M/C = Machine Code.

Press RESET; load the SPS Menu program; load "MCbasldr" by pressing 11 (ENTER). Type in 319 (ENTER) for code length to load. Following the screen instructions, press any key, then PLAY in the FIST Basic Loader. After, press any key (NOT ENTER) to NEW the MCBasldr the FIST loader is still intact at address 32000. To examine this, reload the SPS Menu; type 3 (ENTER) to load "BASCONV". "S" gives screen output. Press "P" for printer output which is automatic here in +3 Basic (a printout is NOT available in 48K Mode). Next, type 23755 (ENTER) for the code Start Address, then type 319 (ENTER) for length of code to analyse. A Basic listing is given which seemed odd. When STOP appears at the bottom of the screen, retry by pressing the "down arrow" key then (ENTER). Type RUN (ENTER) then press "s" for screen display and type in 23813 (ENTER) for the Start Address, then type 319 (ENTER) for the code length to analyse. The Basic listing is now more sensible! This is because the program originated on a computer with a microdrive, which moved PROG up to 23813. Note the following, BORDER 6: PAPER 6: INK 6: CLEAR 24576: PRINT USR 34816. NOTE:- The values shown for CLEAR and PRINT USR were FALSE - the true values were those listed by BASCONV (in brackets). Check for protection thus:- Press RESET and select +3 Basic - type load"t:"(ENTER). Type BORDER 6: PAPER 6: INK 6: CLEAR 24575 (ENTER). Load code "k" using load""code(ENTER). When loaded, type RANDOMIZE USR 34816 (ENTER) - the game ran i.e the code was NOT protected. Conversion to +3 drive is thus just a matter of loading the "k" code with a suitable Basic Loader. Press the RESET button. Select +3 Basic, and type in the following Basic Loader:-

```
10: BORDER 6:PAPER 6:INK 6:CLEAR 24575:LOAD"fist"CODE:PAUSE 100:R:USR 34816 (ENTER)
```

Save to auto run on disk with:- SAVE "FISTLDR" LINE 10 (ENTER). Press RESET. We must now load the game code "k" into the Spectrum's memory from tape, prior to saving it to disc.

Select +3 Basic. Type CLEAR 24575 (ENTER) - keeps the code above RAMTOP. For a load from TAPE not disc, type load"t:" then press ENTER. Type load"code, press ENTER, and PLAY in the "k" game code. When loaded, save the game code to disc with by typing:- save"fist"code 24576,38912 (ENTER). The game can be reloaded with:- load "FISTLDR". We usually compile a menu program of the games on each Disc, thus:-

```
10: BORDER 0: PAPER 0: INK 7: CLS
20: PRINT AT 0,11;"GAMES 1"
30: PRINT AT 1,11;"=====":PRINT:PRINT:PRINT:PRINT
40: PRINT "(1) MANIC MINER.":PRINT:PRINT
50: PRINT "(2) JETSET WILLY.":PRINT:PRINT
60: PRINT "(3) JETPAC.":PRINT:PRINT:PRINT:PRINT
70: PRINT "PRESS (1), (2) OR (3) - (0) TO QUIT."
80: IF INKEY$="" THEN CLS: PRINT AT 10,10;"MANIC MINER":LOAD"MANICMINER"
90: IF INKEY$="2" THEN CLS: PRINT AT 10,10;"JETSET WILLY":LOAD"J.WILLY"
100: IF INKEY$="3" THEN CLS: PRINT AT 10,13;"JETPAC":LOAD"JETPAC"
110: IF INKEY$="0" OR INKEY$="q" THEN RANDOMISE USR 0
120: GOTO 00
```

Save this to auto-run using:-SAVE"disk"LINE 10 (ENTER) - using the name "disk" means that pressing ENTER at the Menu then "1" etc will load that game!

TRANSFERRING PROTECTED MACHINE CODE PROGRAMS.

These are, of course, more difficult, since not only do we have to break the protection, but we have to contend with other complications such as long code blocks, which overwrite the System Variables. We can divide the programs likely to be encountered into 3 main groups e.g.:-

- (1). CODE BLOCKS STARTING AT 25000 OR ABOVE.
- (2). SHORT CODE BLOCKS LOADING BELOW 25000.
- (3). LONG CODE BLOCKS LOADING BELOW 25000.

As stated earlier, we must remember that when we intend saving code blocks with very low Start Addresses, e.g. around 25000, we must realise that if we intend to load or type in any Basic lines, these could overwrite and corrupt the code. Accordingly, we consider that the safest reasonable low address to save is around 25000. This allows for a reasonable amount of Basic and a good safety margin.

We will now consider examples of the 3 cases above to illustrate this. As an example of type (1) above, consider:-

ZYNAPS. (48K).

Load the SP5 Menu and HReader program; PLAY in the tape. This gave:-

Tape Count	Filename	Type	Details
7 - 9	zynaps	Basic	T.L.=81. A.R Line 0.
10 - 14	loader	M/C	S.A.=64512; C.L.=768.
15 - 25	picy	M/C	S.A.=32768; C.L.=4096.
26 - 105	code	M/C	S.A.=23976; C.L.=40535.

NOTE:- A simple (but not infallible) indication of a program being protected is to try and load the Basic Loader using MERGE "" (this stops any auto-run). If the computer crashes or locks up - the program is usually protected. Try using MERGE "" in this case type load"t:" then merge"" and PLAY the tape - the computer locks up!

We must thus load the Basic "zynaps" using the "MChasldr" and "BASCONV" programs as detailed earlier. This gave the following essential details:-

```
10: BORDER 0: PAPER 0: INK 7: CLEAR 32767: LOAD "" CODE: R. USR 64512
```

This showed that the next part (the machine code program "loader") was loaded, then called by R.USR 64512. i.e. it was loaded and called normally. We now know that the "loader" code loads to 64512, is of length 768, and its Execute Address (E.A.) = 64512.

We now need to examine this code. However, if we load it to its normal address of 64512, it will overwrite the KDI disassembler! We thus need to load it to a lower EQUIVALENT address, where it can be examined with KDI. By an equivalent address we mean this:- the S.A. of "loader" = 64512 Decimal = FC00 Hex. An equivalent address here would be 7C00 Hex or 31744 Decimal i.e. the first digit is altered to give a lower address, - the rest stay the same. Press RESET. In +3 Basic type load"t:" (ENTER) then load"code 31744 - press ENTER then PLAY in the "loader". Next, type CLEAR 30000 (ENTER), then NEW (ENTER); load the SP5 Menu, then load KOFTKDI. Examine 7C00 Hex onwards. We can see an unscrambler! This unscrambles the code starting at FC13 Hex. We can use this piece of code to unscramble the code at FC13 by inserting a RETURN instruction at 7C13 (= 31763 Decimal). To do this, return to Basic from KDI, press the "down cursor" key to select +3 Basic. Type clear 30000 (ENTER), NEW (ENTER). Select +3 Basic and type POKE 31763,201 (ENTER) reload the "loader" code by typing load"t:" (ENTER) then load"code (ENTER) then position at start of "loader" and then PLAY it in. Now unscramble the code at 64512 by typing R.USR 31744. Save the code temporarily to disc with:- save"zynaps"code 64531,653. (ENTER). Its Start Address is 64531 = FC13 Hex. - this is the number in the HL register. The code length 653 = 28C is the number in the BC register. Press RESET. Select +3 Basic. Type CLEAR 30000 (ENTER). The equivalent address here is FC13=7C13=31763. Reload the zynaps code from disc using load"a:" (ENTER) then load"zynaps"code 31763 (ENTER). Type NEW (ENTER), reload the SP5 Menu, and reload KDI. Examine 7C13 Hex - it shows JP FE3A - the start address. Also look at 7E3A onwards. This showed:-

At the E.A. of 7E3A Hex (or equivalent address FE3A = 65082) we see a LD SP,FFFF Hex instruction. This sets the stack to the top of memory i.e. 65535. This is similar to a Basic CLEAR 65535. Next, we see DI, or DISABLE INTERRUPTS. This just makes the program run faster. At 7E3E we have a set of BLOCK MOVE instructions, where HL holds the start address; DE holds the destination address; BC holds the number of bytes to move. The LD (HL),0 instruction fills each byte with 0. In short, the effect is to fill the attributes with zero, or make the whole screen black. LDIR at 7E49 executes the block move. The CALL FE74 at 7E4B loads the next block of code (the screen); the CALL 8001 performs a check, the CALL FE74 at 7E51 loads the next code block i.e. "code". The important part starts at address 7E54; this is another block move which moves the 18 bytes from FE62 to FE73 into the printer buffer from 5B00 to 5B12; then jumps to the start at 5B00. At the end is the JP 8000 instruction (at 7E71) - this is the E.A. of the game! Thus, if we POKE addresses FE71, FE72 and FE73 (65137, 65138 and 65139) = 0, the printer buffer addresses 23311, 23312 and 23313 will be = 0. We can then put a piece of "stop" code starting at, say, 23320 to make the program return to Basic after it has loaded the game code, so that this game code can then be saved to tape. This small piece of "stop" code is only 10 bytes long, and can be POKed in. Press RESET, then type CLEAR 50000 (ENTER) - this will ensure the code you are going to load is safe above RANTOP. Reload the zynaps code from disc with load"zynaps"code (ENTER). Type SPECTRUM (ENTER) for 40K Mode, then POKE 65137,0; POKE 65138,0; POKE 65139,0 - this removes the "jump" instruction as stated earlier. We now put the "stop" code at address 23320 - type in the following:-

```
10: FOR A = 23320 TO 23329: INPUT B: POKE A,B: NEXT A
```

RUN it and enter the following numbers i.e. 253,33,58,92,33,3,19,229,251,201. For machine code programmers this gives:-

Numbers	Mnemonics
253,33,58,92	LD IV,5C3A Hex
33,3,19	LD HL,1303 Hex
229	PUSH HL
251	EI
201	RET

NOTE:- These 10 bytes will stop ANY piece of code running, provided the System Variables are not overwritten when the game code is loaded. Next, position the tape at the start of the "picy" file, and type R.USR 65082 (the execute address) and PLAY to load in the game code. After loading, the program will return to Basic - note the usual O.K. message at the bottom of the screen. We now have the game code in memory. Type CLEAR 24998, then save to tape with SAVE"zynaps" CODE 25000, 40536.

How do we know the start address and the length of the code to save? Note that at 7E62 Hex (equivalent to FE62 Hex) there is a block move, where HL=FBFE; DE=FFFE; and BC=9E57. It is an LDDR instruction i.e. the end of the code will be at FFFE, and the start will be FFFE - 9E57 = 61A7 = 24999. The code thus runs from 25000 to 65534. We now need a Basic loader to reload the code:- Press RESET. Select +3 Basic. Type in:-

```
10: CLEAR 24998: LOAD "zynaps" CODE 25000: PAUSE 100: R.USR 32768
```

The CLEAR statement ensures the code is safe above RANTOP, it is loaded then executed by the R.USR statement. The PAUSE 100 gives the disc drive time to stop! Save to disc with: SAVE "ZLDR" LINE 10 i.e. it will auto run on reloading. Type CLEAR 24998; then load"t:"; load the game code from tape with load"code. Type save"a:"; save to disc with save"zynaps"code 25000,40536. The game will reload with LOAD "ZLDR". This method can be adapted for the transfer of other games. Another example of this type is:-

GUNRUNNER. 48K.

Load the Header Reader as shown earlier, PLAY the tape. This gave:-

Tape Count	Filename	Type	Details
12 - 15	Gunrunner	Basic	T.L.=81;P.L.=81; A.R. Line 0
16 - 19	loader	N/C	S.A.=64512; C.L.=768.
20 - 35	picy	N/C	S.A.=32768; C.L.=6912.
36 - 92	code	N/C	S.A.=25319; C.L.=38937.

As in the previous example, trying to load the Basic with MERGE "" gives a crash. So we must load it with MCBasldr and BASCONV as shown earlier. This gave:-

```
10: BORDER 0: PAPER 0: INK 7: CLEAR 25317: LOAD "" CODE: R.USR 64512
```

- does this look familiar? (See "ZYNAPS" method earlier!). The 2 sets of data are virtually identical! This illustrates an important point - the protection methods used by software houses often tend to be the same for many games - it is very expensive to alter! Here, both games are marketed by HEWSON; so if you can master the method, you should find it very similar for other HEWSON games featuring the "countdown" type of loaders. NOTE:- Do not confuse this with the usual countdown games which feature the "ALKATRAZ" system - these give the message "ALKATRAZ PROTECTION SYSTEM" on MERGEing.

Using a similar method as for "ZYNAPS", decode the loader, and reload it to address 31744. Examination with the KD1 disassembler shows a similar pattern; at the execute address of 7E3A Hex (equivalent to FE3A Hex = 65082 Decimal) we see a DI, followed by the attribute block move (7E4B - 7E55). This is followed by the CALL FE76 at 7E48; this

loads the screen display (the "picy" file) to address 32768; it is then moved into the display area by the block move at 7E4B. The CALL FE76 at 7E56 loads the game code and the block move at 7E59 moves 15 bytes into the printer buffer. The E.A. to start the game is the JP BBC6 instruction at 7E73 i.e. the E.A. = BBC6 Hex = 48070 Decimal. Follow a similar method as for "ZYNAPS", but POKE 65139,0; POKE 65140,0; POKE 65141,0; then put the "stop" code at 23320. Type CLEAR 24995; R.USR 65082 and PLAY in the game code. When the program returns to Basic, type CLEAR 26500; then save the game code out to tape with:- SAVE "gun" CODE 26597, 38938. Again, the start and length of the code is obtained from the block move at 7E67 i.e. DE = FFFF; BC = 9818; thus the code ends at FFFF Hex, and starts at (FFFF - 9818) = 67E7 Hex = 26599 i.e. the code runs from 26599 to 65335. Press reset, select +3 Basic. Type load "t:", then CLEAR 26500. Load the game code from tape with LOAD "gun" CODE. Type save "a:"; save the code to disc with SAVE "gun" CODE 26597, 38938. Type in the following Basic Loader:-

```
10: BORDER 0:PAPER 0:INK 7:CLEAR 26500:LOAD"gun"CODE:PAUSE 100:RANDOMIZE USR 48070
Save to disc to auto run from line 10 with:- SAVE"GR" LINE 10. The game will reload
with:- LOAD"GR".
```

An example of type (2) i.e. a short block loading below 25000 is:-

THE LIVING DAYLIGHTS. (48K).

Again, using the Header Reader program, we find:-

Tape Count	Filename	Type	Details
0 - 6	Bond	Basic	T.L.=380; P.L.=373; A.R. Line 0.
7 -->	Headerless blocks		

Loading the Basic loader "Bond" using "MCbasldr" and "BASCONV" gave for the important points:-

```
0: R.USR 23768
```

- the E.A.! Note that sometimes you must enter the start of the code when analysing a Basic program as 23755; other times you must enter 23813; this is dependent on whether or not a microdrive was present when the program was originally saved. Here, enter 23813

Note the REM statement followed by all the apparently meaningless tokens - this is machine code contained in the REM statement, as shown earlier! This is the machine code we are after to load the game. Press the "down" cursor key to select +3 Basic. Type CLEAR 30000 then NEW. Reload the SP5 Menu and load KSFTKD1 by pressing key 9 then ENTER. NOTE:- The R.USR value of 23768 = PROG + 13 for an unexpanded Spectrum. Thus, examine the area from 32013 (i.e. 7D00 Hex) onwards with KDI. This is because the MCbasldr program loads the Basic to address 32000.

Using KDI, you will see at 7D00 a block move which effectively moves the whole screen along one byte. This seems to suggest some sort of screen check may be employed, so it will be wise to save the screen area as well as the main code! At 7D1B there is another block move which moves the area from 7D29 on up to FE00 i.e. 65024, then jumps to this address. The code from 7D29 on is thus the code to load the game. At 7D29 we see the instruction LD IX,4000 - this is the start address of the code = 16384 Decimal. At 7D2D we see LD DE,9600 - this is the code length = 30400 Decimal. The game code thus runs from 16384 to 54784. This gives us another problem - it will overwrite all the System Variables and Basic areas, making a simple return to Basic impossible! The only solution is the one normally employed in this case; load the code to a higher address originally, then save it out with a piece of "mover" code tagged on the end which will enable the code to be reloaded to its high position, then moved back down to its normal position, and jump to the Execute Address; WITHOUT RETURNING TO BASIC. To do this, we must alter the value in IX. A convenient value here would be, say, 26000. The code would then load from 26000 to 64400. This will then let us return to Basic to save it. This is done by returning to Basic from KDI, press the "down" cursor key, in +3 Basic type poke 32043,144; poke 32044,101; this puts the address (101*256)+144 = 26000 in IX at 7D2B. We must also change the area moved at 7D1B (which = 5CF4), since this relates to the NORMAL position of the Basic. The address 5CF4 must be changed to 7D29. This can be done by typing POKE 32028,41; POKE 32029,125. We now need to find the E.A. Looking at address 7D98 we see JP 9100 i.e. 37120 - the E.A.! To force a return to Basic here instead of the jump to start the game, type POKE 32152,251 followed by POKE 32153, 201. This gives EI (enable interrupts), then RET (return to Basic after loading the game code). Position the tape after the Basic loader "Bond", type CLEAR 25990 (keeps the code above RAMTOP), then R.USR 32027 and PLAY in the game code. Now we must add the piece of "mover" code to the end of the game code. To do this, enter the following:-

```
1: FOR A = 64400 TO 64410: INPUT B: POKE A,B: NEXT A:STOP
```

RUN it and type in these numbers:- 33,144,101,17,0,64,1,20,150,237,176,195,0,145.
For machine code programmers, this gives:-

```
LD HL, 26000
LD DE, 16384
LD BC, 38420
LDIR
JP 37120
```

or "move the 38420 bytes starting at 26000 to 16384, then jump to the E.A. of 37120. Save this together with the game code to disc with SAVE "bond" CODE 26000, 38420. Press RESET, select +3 Basic; and type in the following Basic loader:-

10: CLEAR 65535: LOAD "bond" CODE 26000: PAUSE 100: R.USR 64405

Save to disc to auto run from Line 10 with SAVE "LD" LINE 10. The game reloads with: LOAD "LD". Note:- The CLEAR 65535 command must be included to avoid the stack being overwritten when the code is moved back down. The R.USR is 64405 since this is the E.A. to start the move down, and jump to the start of the game.

An example of type (3) is:-

SUPER SPRINT. (48K).

These long blocks loading below 25000 are by far the most difficult to transfer to +3 Drive. Applying the normal method, load the Header Reader program. This gave:-

Tape Count	Filename	Type	Details
5 - 7	Super	Basic	T.L.=201;P.L.=201;A.R. Line 10.
8 - 11	load	M/C	S.A.=32768; C.L.=448.
15 --->	Fast Loading Code		

Following the usual method, load the Basic part "Super" using MCBASLD and BASCONV. This gave, as the most important points:-

```
10: PAPER 0: INK 0: BORDER 0: CLEAR 32767
20: LOAD "" CODE 65088
30: RANDOMIZE USR 65088
```

i.e. the program "load" is the machine code to load the game. We now need to load the "load" code to a lower equivalent position to examine it with the KDI disassembler (see example "ZYNAPS" earlier). The code normally loads to 65088 = FE40 Hex. A suitable lower EQUIVALENT address is thus 7E40 Hex = 32320 Decimal. Press the RESET button. Select +3 Basic, type CLEAR 32000, then load "t:", then load the code with load "" code 32320. Type NEW, then load the SP5 Menu, and then press key 9 to load the KSFTKDI disassembler - and examine the code starting at 7E40 Hex. From 7E40 to 7E53 we see an unscrambler! The unscrambling starts at FE53 (equivalent to 7E53) - the address in the HL register at 7E40. NOTE:- This will alter the last byte of the JR instruction at 7E53! However, this type of unscrambler always repeats the LD A,R instruction at 7E4A; this is thus the correct JR point. Thus, we must alter the byte at 7E53 from F4 to F6. Press RESET. In +3 Basic, type CLEAR 32000, then load the code using load "" code 65088, then load "" code 32320. Type POKE 32339,246; POKE 32340,251; POKE 32341,201. This puts

the instructions EI (enable interrupts), and RET (return), at addresses 7E54 and 7E55, which makes the program return to Basic after it has unscrambled the code at 65088. Now, activate the unscrambler with R.USR 32320. The code at 65088 has now been unscrambled and is saved to disc with SAVE "u" CODE 65088,448. RESET. In +3 Basic, type CLEAR 32000, and reload from disc with LOAD "u" CODE 32320. Type NEW. Load SP5 Menu, then KDI and disassemble from 7E40 Hex. The end of the JR of the unscrambler is seen at 7E53, the program then falls through to the next instruction below, which is JP FF2C at 7E54. This is the start address to load the game code. Now we need to make the program load the game code, then return to Basic to save it to tape. Looking at 7F2C we see the instruction LD SP,5D80 which sets the Stack to address 5D80 or 23984 Decimal. At address 7F3D we see the instruction LD IX,4000; then LD DE,1800. Remember, IX holds the Start Address of a block of code, DE holds its length. Here, the S.A. is 4000 Hex or 16384 Decimal; C.L.=1800 Hex or 6912 Decimal. This is the code to load the picture. Also we see:- At 7F47 S.A.=23989; at 7F48 C.L.=36864. At 7F51 S.A.=60853; at 7F55 C.L.=4170. Thus, the code is in 2 blocks and runs from 23989 to 65023. At 7F5B we see a block move of the code from FF69 to 5D80, then a JUMP to 5D80. We must thus include the code from FF69 to FF7A in any code we use to modify the program. A good place to put the "stop" code is 7F5B or the equivalent FF5B=65371. Press the RESET button, and then in +3 Basic we type CLEAR 32000 and then reload the code from disc with LOAD "u" CODE 65088. Type POKE 65107,243. This puts a DI (disable interrupts) instruction at the start address we will use i.e. FE53 Hex = 65107 Decimal. We will now enter the "stop" code. Type:-

```
10: FOR A=65371 TO 65303:INPUT B:POKE A,B:NEXT A
```

RUN it and enter the following:- 0,0,0,253,33,58,92,33,3,19,229,251,201. For machine code programmers, this is:-

```
LD SP,FFFF
LD IX,5C3A
LD HL,1303
PUSH HL
EI
RET
```

This will stop the program after loading and return to Basic. Type CLEAR 65535. Since the S.A. of the first code block is so low (23989), delete line 10 of the above Basic by typing 10 ENTER. Type POKE 65324,0; POKE 65325,0; POKE 65326,0. Type R.USR 65107 - PLAY in the game starting from the first headerless block. On returning to Basic, save the first 2000 bytes to disc with SAVE "a" CODE 23989,2000. Save the last part with SAVE "b" CODE 25989,39834. Press RESET. The next move is to reload the "a" code to a suitable address e.g. 32000, and add a piece of code to move it back to its usual place after it is reloaded to the screen at 16384. Load the code with LOAD "a" CODE 32000. Now type in:-

10: FOR A=34002 TO 34033:INPUT B:POKE A,B:NEXT A

RUN it and enter the following numbers:-

243,49,176,93,33,0,64,17,181,93,1,208,7,237,176,33,0,254,17,0,0,1,80,160,237,184,175,211,254,195,0,154. For machine code programmers, this is:-

```
DI
LD SP,5DB0
LD HL,4000
LD DE,5DB5
LD BC,07D0
LDIR
LD HL,FE00
LD DE,0000
LD BC,A050
LDDR
XOR A
OUT (FE),A
JP 9A00
```

The first half down to the LDIR moves the code from the screen into place. The second part is the code from the original program. Save to disc with SAVE*c*CODE 32000,2034. We have now saved all the necessary code, and now need to transfer to +3 Drive. Press RESET. Select +3 Basic. Type in the following Basic Loader:-

10: CLEAR 25000: LOAD*b*CODE: LOAD*c*CODE 16384:PAUSE 100:RANDOMIZE USR 18386

NOTE:- The S.A. is calculated thus:- it is (34002-32000) = 2002 bytes from the start i.e. 16384 + 2002 = 18386. Save to disc to auto-run from Line 10 with SAVE*SS*Line 10. The game reloads with LOAD*SS*.

Using modifications on the above methods, other programs may be transferred to your +3 Drive.

"SPEEDLOCK DECODER SD1"

INSTRUCTIONS

INTRODUCTION.

One form of software protection which has recently been developed is the famous (or infamous) SPEEDLOCK system - which uses the dreaded "Pulsed Leaders" to load the program. These have lead to many loading problems, and because of this we have produced our "SD1 SPEEDLOCK DECODER" system. We did this since these programs are now so widespread - being used by several software houses including Ocean, US Gold, and Imagine to name but a few.

SD1 converts the pulsed leaders to normal ones, and saves the program out at normal speed. It also saves the program code in two well defined blocks, with NORMAL headers, and gives the start address for the machine code, as well as producing a Basic loader to reload the program for Tape (if required), and code for +3 Drive conversions. With SD1 and SD1 Trans these programs are now easily transferred to disc. As mentioned above, we have included a new program "SD1 Trans" which easily transfers the code blocks saved on tape by SD1 to disc. The SD1 Trans program is loaded, then the tape obtained using SD1 is simply PLAYed in - transfer to disc is then simply a matter of stopping and starting the tape according to the instructions displayed on the screen.

HOW TO TELL WHICH SPEEDLOCK SYSTEM A PROGRAM CONTAINS.

The OLDER type Speedlock games had 1 (sometimes 2) Basic parts, these were usually fairly short, and were followed by the "Pulsing Leaders". Here, instead of seeing the usual RED and CYAN (light blue) loading stripes in the border, we see "jerking" stripes which load with a "clicking" sound. Examples are "BATMAN", "WINTER GAMES", "HEAD OVER HEELS" and "IMPOSSIBLE MISSION". The colour of the stripes was later changed to RED and BLACK on games such as "ENDURO RACER", "ARKANOID" - all these are OLD type Speedlocks, and SD1 will convert these. The NEWER Speedlock games are completely different. Speedlock Type 2 programs have a very short then a long Basic part. This is followed by the border turning red/black and a series of random musical "beeps" is heard. Following this is a very short red/black part; then the start of a long block loading with a blue/black border. Also, there is an on screen counter e.g "WIZBALL". SD234 transfers these EASILY to disc; see later. Type 3 Speedlocks are as for Type 2, but with "clicks" not "beeps"; SD234 handles these. The NEW TYPE 4 Speedlocks have a LONG Basic loader, then a short part loading with a blue/black border followed by a long block loading with a blue/black border and an on screen counter; SD234 transfers these VERY EASILY (see later). Examples of games using Speedlock 4 are:- "FIREFLY", "GUTZ", and "ARKANOID 2". Type 5 Speedlocks, as used in SD5 are similar. Examples of these are "BATMAN2", "DRAGON NINJA", "ROBOCOP" and "R-TYPE". PLEASE NOTE:- SD234 and SD5 WILL transfer "Multi-Load" programs, but only the long first part is transferred. No hardware device can transfer these either, as the loader code is in the program, which cannot be changed.

LOADING SD1.

Firstly, make sure you clear your Plus 3 by pressing the RESET switch on the side. Next, insert the SP5 disc, and at the +3 opening Menu, press ENTER - the SP5 Menu will load and run. To load SD1, press key 14 then ENTER, when SD1 will load and run automatically, displaying the message:- "Press ENTER then PLAY Tape".

(1). TRANSFERRING OLD TYPE SPEEDLOCK PROGRAMS.

When the message "Press ENTER then PLAY Tape" appears, insert your Speedlock program tape, and rewind it to the start. SD1 is now ready to use.

Some old Speedlock programs have ONE Basic section, others have TWO, before the pulsing sections. Where TWO Basic sections are present, SD1 will ignore the first. Press ENTER, then PLAY on your recorder. When SD1 sees the main Basic, it will display its name, which will be used when saving Basic and code blocks. Watch your screen, and when the main Basic has loaded you will hear a warning beep and see a message saying "Stop the Tape". STOP your recorder - there will be a short pause while SD1 decodes the Speedlock Basic, then the message "Start Address of Code = *****" will appear. This is the E.A. (RUSR number) of the converted program for +3 Drive conversion - write it down. This is entered in the SD1 Trans program (see later).

NOTE:- You may sometimes see the message:- "Decoding Error - Press ENTER". If this happens, rewind your Speedlock tape to its start, press ENTER, then PLAY to try again. If the error reoccurs, try again using a slightly different volume, if it still happens, assume the program isn't Speedlock, or SD1 can't convert it. The Menu available is:-

Key	Function
"1"	- start loading a NEW Speedlock program.
"t"	- Save Basic loader for TAPE use.
"c"	- Save Code only for +3 Drive conversion.
"n"	- Next stage; loading pulsing parts.
BREAK	- Performs a reset or NEW.

Now, if you wish to go back to the start, simply press the "1" key, otherwise place a new tape into your recorder, start recording, and according to which system you are converting for, press the appropriate key i.e.:- the "t" or the "c" key. Press "t" for a copy that is to be reloaded from tape. Press "c" if you are transferring to +3 Drive. ENSURE you press RECORD and PLAY on your recorder BEFORE pressing a key, since pressing a key saves out to tape immediately. On pressing the appropriate key, you will save a Basic program that will load in the rest of the program's machine code i.e. a Basic Loader. This will contain the equivalent of LOAD"a"CODE for tape.

NOTE:- +3 Drive owners must press "c" for a transfer to the +3 Drive.

Having now saved a program to tape, we go on to the next section:-

Having saved out to tape, replace the Speedlock tape in your recorder, and press the "n" key. This will show the FINAL Menu, i.e.:-

- (1) Save Code (Tape or +3 Drive).
- (5) Start Game.
- BREAK = NEW

NOTE:- This is the Menu available AFTER all pulsing sections have been loaded. Start loading all the pulsing sections by pressing ENTER, followed by PLAY on your recorder. Wait until SD1 has loaded in ALL the program - this is shown by the border going YELLOW. If the screen display is corrupted, this indicates a loading error - this means you must RESET the Spectrum and reload SD1.

The keys available now are as described above i.e. (1), (5) or BREAK.

Replace your saving tape into your recorder and set ready to record. For Tape, and / or +3 Drive transfer, press RECORD and PLAY, followed by key (1).

Wait until the program has been saved (border YELLOW). For tape backups the process is now complete. The program reloads with the usual LOAD "" ENTER command. When the program has loaded, (screen loaded), Press ENTER to start the game.

NOTE:- Pressing "c" for +3 transfer results in THREE blocks of code being saved to tape. These can then be transferred to disc EASILY using the new SD1 Trans program provided (see below).

NOTE:- After saving, you can either reset (clear) the Spectrum by pressing BREAK, or play the game by pressing key (5).

FINAL TRANSFER TO +3 DRIVE.

To complete the process, we now transfer the 3 blocks of code on the SD1 tape to your disc using the "SD1 Trans" program provided. Press RESET. Rewind the SD1 tape obtained above back to the start. Insert the SP5 disc, then press ENTER. At the SP5 Menu 15 ENTER loads the "SD1 Trans". The message "Filename? (5 Letters)= " appears. Enter the name you choose (FIVE letters ONLY please) and press ENTER. The message "Start Address =" appears. Type in the number you wrote down earlier and press ENTER. The messages "Put a disk in the Drive" and "Press any key to save Loader" appear. Insert a formatted disc in the drive, and press any key - a Basic Loader will be saved out to disc. The screen will clear and the message "PLAY the Tape" then appears. Continue PLAYing and STOPping the tape as displayed on the screen - the code blocks are automatically saved for you! At the end, the message "Transfer to Disc Complete" appears. Also "Filename is:- name", where "name" is the name used to reload the program from disc.

"SPEEDLOCK DECODER SD234"INSTRUCTIONS(2). TRANSFERRING TYPES 2,3 AND 4 SPEEDLOCKS.

For this we use the new SD234 program. We must first transfer program "Slockldr" from the SP5 disc to the disc on which you are to save your Speedlock programs. Load the SP5 menu. At the Menu, press 16 then ENTER to load "Slockldr". Ignore the error message "Nonsense in Basic". Remove your SP5 disc, and insert a formatted disc on which you wish to save your Speedlock programs. Press "down" cursor, then ENTER for +3 Basic; type save"disk"line 1 (ENTER) and the program will be saved to disc. PLEASE NOTE:- This is a special loader program and as such, it MUST only be placed on a disc ONCE, and no other program called "disk" must be present on that disc. Remove your Speedlock disc; insert the SP5 disc. We now load SD234. Press RESET. Load the SP5 Menu, press 6 ENTER and SD234 will load and then run. When loaded, it waits for you to load the Speedlock game you wish to transfer to disc. NOTE:- The screen remains unchanged when SD234 has loaded, except for a small line in the top L.H. corner. Remove your SP5 disc; insert the Speedlock disc for saving.

Simply PLAY the game in from the start until it has all loaded. SD234 will then ask you to enter the name for your program. Type in a name which can be up to ELEVEN letters and press ENTER. Then press the "D" key for a save to disc. NOTE:- SD234 compresses the screen, which becomes jumbled, but this is normal. The program is RAPIDLY saved out to disc. To reload, press RESET. At the Menu, press ENTER with the disc in the drive. The loader runs and displays a Menu of the programs on the disc. e.g. (1) ARKANOID (2) DISK (3) FIREFLY etc. PLEASE NOTE:- We have included a novel new feature in this loader for our SD234 decoder. It is as follows. Some games, especially Speedlock 2 and 3 games, lock the +3 in 48K mode on loading, thus only the limited 48K sound is available. However, if when loading a game using the new loader, you hold down a key WHILE THE PROGRAM IS LOADING FROM DISC, the +3 will be left in 128K mode and the FULL sound will be available. NOTE:- ANY key can be held down, but make sure it is NOT a key which is used in the Menu at the start of the game! Key "z" is a fairly safe bet. The effect is vividly shown by transferring "WIZBALL" to disc. Reload without holding down a key, and compare this sound to that obtained when it is reloaded whilst depressing a key. This effect is not available even with some hardware transfer devices! Load by pressing and/or holding the key i.e. "1", "2" or "3". The game loads and runs usually within a few SECONDS! What could be easier? Remember, SD234 will work for types 2,3, and 4 Speedlock programs - see earlier for how to identify them. For type 1 programs, use SD1.

"SPEEDLOCK DECODER SD5"INSTRUCTIONS(3). TRANSFERRING TYPE 5 SPEEDLOCKS.

For this we use the new SD5 program. We must first transfer program "SDSLDR" from the SP5 disc to the disc on which you are to save your Speedlock programs. Load the SP5 Menu. At the Menu, press 18 then ENTER to load "SDSLDR". Ignore the error message "Nonsense in Basic". Remove your SP5 disc, and insert a formatted disc on which you wish to save your Speedlock programs. Press "down" cursor, then ENTER for +3 Basic; type save"disk"line 1 (ENTER) and the program will be saved to disc. PLEASE NOTE:- This is a special loader program and as such, it MUST only be placed on a disc ONCE, and no other program called "disk" must be present on that disc. Remove your Speedlock disc; insert the SP5 disc. We now load SD5. Press RESET. Load the SP5 Menu, press 17 ENTER and SD5 will load and then run. When loaded, it waits for you to load the Speedlock game you wish to transfer to disc. NOTE:- The screen remains unchanged when SD5 has loaded, except for a small line in the top L.H. corner. Leave your SP5 disc in the drive for the time being.

Simply PLAY the game in from the start with the volume turned up on your TV set or monitor. When the main part of the Speedlock at the start has loaded, you will hear a load beeping sound and the border on your screen will flash violently - STOP THE TAPE at this point. As mentioned earlier, the SD5 program contains a software "emulator" program which will decode the Speedlock in the game. This is signified by the flashing border and the buzzing sound. As the decoding proceeds these effects get faster and faster until they stop - the Speedlock is then decoded. This method takes a little longer than the SD234 method, but it will decode far more programs, making it more flexible in use. When the buzzing and flashing have finished, press PLAY on your recorder and continue PLAYING the tape.

When finished, SD5 will then ask you to enter the "PASSWORD DISK". This is a protection method. Simply press ENTER (your SP5 disc should be in the drive). If the SP5 disc is a copy, the program detects this and will proceed no further. With the original SP5 disc, the message "INPUT FILENAME" appears. Type in a name, which may be up to ELEVEN letters then press ENTER. The next message is "D=SAVE TO DISC". Remove your SP5 disc, and insert the disc for your Speedlock programs. Press the "D" key and the program will be saved to disc. NOTE:- SD5 also compresses and jumbles the screen - this again is normal. To reload games from your Speedlock disc, press RESET, then at the +3 Menu press ENTER. The Loader on the disc will give a Menu i.e. (1) BATMAN2 (2) ROBOCOP (3) DISK (4) RTYPE etc. Press the appropriate key i.e. the game of your choice. It usually loads in around 5 SECONDS! SD5 will enable you to transfer most Speedlock programs.

"ALCATRAZ DECODER AD1"**INSTRUCTIONS****INTRODUCTION.**

Another popular form of protection system which has recently been evolved is the "ALCATRAZ PROTECTION SYSTEM". It is used by software houses such as Ocean, US Gold and The Edge. Its presence is shown by a loading sequence which has no changing border colours; a screen display which loads in an unusual way; and a 3 digit counter which reduces to zero. To confirm, press RESET, select +3 Basic, type load"t:" press ENTER, then type MERGE "" ENTER; PLAY in the Basic. LIST shows "ALCATRAZ PROTECTION SYSTEM".

Again, the problem with these programs are they are VERY sensitive to loading volume changes. Thus, it is a good idea to ensure before trying to convert these programs, that the correct volume has been selected to load the original!

We have thus evolved our "AD1 ALCATRAZ DECODER", which converts the programs to standard format regarding speed and headers. This enables easy transfer to tape, (if required), and to your +3 Drive, in conjunction with our "AD1 Trans" program.

LOADING AD1.

Firstly, ensure that your +3 is cleared by pressing RESET. Insert your SP5 disc, then press ENTER. The SP5 Menu program will load automatically.

To load AD1, at the SP5 Menu, press key 1 then press the ENTER key - AD1 will load and run displaying the message:- "Press ENTER then PLAY Tape". When you see this message insert the Alkatraz tape into your recorder, and then rewind it to the start. AD1 is now ready to use.

Press ENTER, then PLAY on your recorder. Watch your screen, when the main Basic has loaded you will hear a warning beep and see a message saying "Stop the Tape. STOP your recorder - there will be a short pause while AD1 decodes the Basic, then you can select a Basic loader (this will be for TAPE only) by pressing "t". You may see the message:- "Tape Error - Please press ENTER". If this occurs, press ENTER, rewind the tape, and repeat, trying a slightly different volume. If this persists, either the program is not an Alkatraz program, or AD1 can't convert it. Remove the Alkatraz tape, and insert a new tape, press RECORD and PLAY, and press "t" to save Tape Basic. For a transfer to +3 Drive - ignore this step and proceed as below.

Press "n" to go to the next stage. Replace the Alkatraz tape, press ENTER, then PLAY. When the next part has loaded, STOP the tape and AD1 will decode this part. Press ENTER then PLAY again. AD1 will load in the rest of the program as normal. When successfully loaded, the screen will appear jumbled with a YELLOW border. If an error occurs, (no jumbled screen or yellow border), reload AD1 and start again.

On successful loading, reinsert your saving tape, press RECORD and PLAY, then press key (1). This applies irrespective of whether you are saving for tape or for +3.

AD1 will then save 3 blocks of code with the original name, but with the tenth character altered to 1,2 and 3. Repeat copies can be obtained by pressing key (1), or the Spectrum can be cleared by pressing BREAK. Alternatively, you can play the game by pressing key (5), then ENTER. A TAPE conversion is completed at this point. You now have a Basic loader and three pieces of code on tape for a TAPE conversion. For a transfer to +3 Drive, you will have just 3 pieces of code on tape.

FINAL TRANSFER TO +3 DRIVE.

The three pieces of code can now be easily transferred to disc using the new "AD1 Trans" program and "Loader" program supplied with SP5. Firstly, we must transfer the "Loader" program to disc. This it does itself thus:- Press RESET. Load the SP5 Menu, then press key 1B and ENTER. The "Loader" program will load and run. It simply waits for you to press a key and it transfers itself to a disc. Thus, when loaded, put a formatted disc in the drive, and press any key. This will save a loader program to disc. NOTE:- This needs to be done only ONCE for each side of the disc. Press RESET. Remove the disc. Reinsert your SP5 disc, and load the SP5 Menu, then press key 2 and ENTER to load the "AD1 Trans" program. Remove the SP5 disc, insert the disc with the saved loader. When "AD1 Trans" is loaded "Filename? (5 letters)" appears. Type in 5 letters for the name of the program you wish to save. Press ENTER. The message "PLAY the Tape" appears. Insert the AD1 tape obtained as detailed above, rewind it, and press PLAY. Follow the on screen instructions to PLAY and STOP the tape when necessary. The code blocks are saved to disc automatically. To reload a program, press RESET. At the +3 Menu, press ENTER. The loader will run and ask you to enter the name of the program you wish to enter. Type in the name, press ENTER, and the program will load and run. After loading, when the screen display appears, press ENTER, and the game will run.

Remember, to save time, always check first to see if the game you intend transferring will load from tape in +3 Basic mode, since some Alkatraz programs are incompatible with +3 mode.

Examples of games protected by the Alkatraz system are "BOBBY BEARING", "FAIRLIGHT", "RYGAR", "SUPER CYCLE" and "BREAKTHRU".

Using a similar method, other Alkatraz games may be transferred to your drive.

"ALKATRAZ DECODER AD2"INSTRUCTIONSINTRODUCTION.

Our new AD2 program enables the EASY transfer of the VERY LATEST Alkatraz programs using the following procedures:-

- (1). Load AD2, then PLAY in the Alkatraz tape.
- (2). When loaded, press ENTER, which saves 3 code blocks to a new tape.
- (3). Load the transfer program TRANSFER; put a formatted disk in your +3 drive then PLAY in the saved tape - after loading, the program is automatically transferred to your disk, and reloads by pressing key ENTER at the +3 Menu.

The FULL details are now given below:-

LOADING AD2.

First RESET your +3. At the Menu, press ENTER to load the SP5 Menu, then press keys 19 then ENTER, and AD2 will load and run.

PLEASE NOTE:- BEFORE loading AD2 you must ALWAYS RESET your Spectrum to clear the memory. When AD2 has loaded, the message "PLAY THE TAPE" appears. Leave your SP5 disc in the drive; put the tape for transfer into your tape recorder. Rewind it to the start and press PLAY. When the first long part has loaded, the border will flash as AD2 decodes the Alkatraz, and a few coloured squares appear toward the top of the screen, but this is normal. Now leave the tape PLAYING. The program should load as normal. When loaded - this is usually when the on screen counter reaches zero - remove your game tape and insert a new tape. Press REC + PLAY to save then press ENTER - 3 headerless blocks will be saved to your tape. On completion of saving, the game will run as normal - play it if you wish. Rewind the saving tape. RESET the Spectrum, and load the "TRANSFER" program from your SP5 disc by pressing keys 28 then press ENTER. When loaded, the screen goes BLACK and the Spectrum goes into tape loading mode. Put a formatted 3" disc in the disc drive. Insert your saved tape and press PLAY. TRANSFER will load the 3 blocks.

NOTE:- When the blocks are loading some distortion of the upper part of the screen will appear, but this is normal. When the tape has loaded, TRANSFER will ask you to enter the name you wish to call the program by displaying the message "Name?". Enter the name you choose, (up to SEVEN letters ONLY) e.g. "BLADE", and press ENTER. TRANSFER will then save the code blocks to your disc as "BLADE1" and "BLADE2" i.e. the original name with the suffix 1 and 2. It then asks "Save DISK?". This is a loader program to reload your programs and there MUST ONLY BE ONE SUCH DISK FILE ON EACH SIDE OF THE DISK. To save "disk" press "y" (ENTER) or "n" (ENTER) to ignore. The +3 then resets. To reload a program, RESET the +3. At the +3 Menu, press ENTER. This loads the "disk" program, which will also give a list of the programs on the disc, and asks "Load?". Type in the name you chose i.e. BLADE and press ENTER when it will load and run.

"ALKATRAZ DECODER AD3"INSTRUCTIONSINTRODUCTION.

Our new AD3 program enables the EASY transfer of the VERY LATEST Alkatraz programs using the following procedures:-

- (1). Load AD3, then PLAY in the Alkatraz tape.
- (2). When loaded, press BREAK, which saves 3 code blocks to a new tape.
- (3). Load the transfer program TRANSFER; put a formatted disk in your +3 drive then PLAY in the saved tape - after loading, the program is automatically transferred to your disk, and reloads by pressing key ENTER at the +3 Menu.

The FULL details are now given below:-

LOADING AD3.

First RESET your +3. At the Menu, press ENTER to load the SP5 Menu, then press keys 22 then ENTER, and AD3 will load and run.

PLEASE NOTE:- BEFORE loading AD3 you must ALWAYS RESET your Spectrum to clear the memory. When AD3 has loaded, the message "PLAY THE TAPE" appears. Leave your SP5 disc in the drive; put the tape for transfer into your tape recorder. Rewind it to the start and press PLAY. When the first long part has loaded, the border will flash as AD3 decodes the Alkatraz, and a few coloured squares appear toward the top of the screen, but this is normal. Now leave the tape PLAYING. The program should load as normal. When loaded - this is usually when the on screen counter reaches zero - remove your game tape and insert a new tape. Press REC + PLAY to save then press BREAK - 3 headerless blocks will be saved to your tape. On completion of saving, the game will run as normal - play it if you wish. Rewind the saving tape. RESET the Spectrum, and load the "TRANSFER" program from your SP5 disc by pressing keys 28 then press ENTER. When loaded, the screen goes BLACK and the Spectrum goes into tape loading mode. Put a formatted 3" disc in the disc drive. Insert your saved tape and press PLAY. TRANSFER will load the 3 blocks.

NOTE:- When the blocks are loading some distortion of the upper part of the screen will appear, but this is normal. When the tape has loaded, TRANSFER will ask you to enter the name you wish to call the program by displaying the message "Name?". Enter the name you choose, (up to SEVEN letters ONLY) e.g. "BLADE", and press ENTER. TRANSFER will then save the code blocks to your disc as "BLADE1" and "BLADE2" i.e. the original name with the suffix 1 and 2. It then asks "Save DISK?". This is a loader program to reload your programs and there MUST ONLY BE ONE SUCH DISK FILE ON EACH SIDE OF THE DISK. To save "disk" press "y" (ENTER) or "n" (ENTER) to ignore. The +3 then resets. To reload a program, RESET the +3. At the +3 Menu, press ENTER. This loads the "disk" program, which will also give a list of the programs on the disc, and asks "Load?". Type in the name you chose i.e. BLADE and press ENTER when it will load and run.

"FIREBIRD DECODER FB1"INSTRUCTIONSINTRODUCTION.

Our new FB1 program enables the EASY transfer of those slow loading "multiple small block" programs from "FIREBIRD", using the following procedure:-

- (1). Load FB1, then PLAY in the Firebird tape.
- (2). When loaded, press BREAK, which saves 3 code blocks to a new tape.
- (3). Load the transfer program TRANSFER; put a formatted disk in your +3 drive then PLAY in the saved tape - after loading, the program is automatically transferred to your disk, and reloads by pressing key ENTER at the +3 Menu.

The FULL details are now given below:-

LOADING FB1.

First RESET your +3. At the Menu, press ENTER to load the SP5 Menu, then press keys 23 then ENTER, and FB1 will load.

PLEASE NOTE:- BEFORE loading FB1 you must ALWAYS RESET your Spectrum to clear the memory. When FB1 has loaded, a STOP message will appear at the bottom of the screen. Next, press the "down arrow" key then ENTER to go into +3 Basic. Next, type SPECTRUM and press ENTER to go into 48K Basic. Type RUN (press the "R" key) then press the "3" key so you get the line RUN 3 then press ENTER. The disc will run for a short time, then the message "PLAY THE TAPE" will appear. FB1 is now ready to use. Put the tape for transfer into your tape recorder. Rewind it to the start and press PLAY. Continue PLAYING the tape until it has completely loaded. When the loading has finished, (Loading counter reaches a maximum then stops), remove the Firebird tape and insert a new one. Press REC+PLAY to save then press BREAK - 3 headerless blocks will be saved to your tape. On completion of saving, the game will run as normal - play it if you wish. Rewind the saving tape. RESET the Spectrum, and load the "TRANSFER" program from your SP5 disc by pressing keys 28 then press ENTER. When loaded, the screen goes BLACK and the Spectrum goes into tape loading mode. Put a formatted 3" disc in the disc drive. Insert your saved tape and press PLAY. TRANSFER will load the 3 blocks. NOTE:- When the blocks are loading some distortion of the upper part of the screen will appear, but this is normal. When the tape has loaded, TRANSFER will ask you to enter the name you wish to call the program by displaying the message "Name?". Enter the name you choose, (up to SEVEN letters ONLY) e.g. "SHARK", and press ENTER. TRANSFER will then save the code blocks to your disc as "SHARK1" and "SHARK2" i.e. the original name with the suffix 1 and 2. It then asks "Save DISK?". This is a loader program to reload your programs and there MUST ONLY BE ONE SUCH DISK FILE ON EACH SIDE OF THE DISK. To save "disk" press "y" (ENTER) or "n" (ENTER) to ignore. The +3 resets. To reload a program, RESET the +3 and at the +3 Menu press ENTER. This loads the "disk" program, which will also give a list of the programs on the disc, and asks "Load?". Type in the name you chose i.e SHARK and press ENTER when it will load and run.

"DISKCAT DISC CATALOGUER"INSTRUCTIONSINTRODUCTION.

With our NEW "DISKCAT" disc catalogue program, you can now get a FULL Catalogue of ALL the files on your discs QUICKLY and efficiently. "DISKCAT" also gives you many more facilities which make it a superb utility in its own right, AND AT NO EXTRA COST!

LOADING DISKCAT.

RESET your Spectrum. Insert your SP5 disc and at the +3 Menu, press ENTER to load the SP5 Menu. Press keys 21 then ENTER, and "DISKCAT" will load and run, displaying the message "INSERT DISK-PRESS ANY KEY TO CAT". Insert the disc you wish to Catalogue and press any key. The files will be listed on the screen. Each page of files is shows a maximum of 28 files. For subsequent pages, press the "N" key. This gives the catalogue of ALL files on the disc. The various columns give, from left to right:-

FILE NUMBER:- These are used in the various sub menus below.

FILENAME and EXTENSION (if any):- The full filename of up to eight letters is given, plus any extension after the full stop.

START ADDRESS:- For Basic this gives the auto run line number; for code the start address in memory.

LENGTH:- For Basic this is the full program length; for code the total block length.

TYPE:- The file type. B=Basic; C=Code; H=Unknown format.

The keys available in the Menu are:- N,A,E,U,P,S,R,D,F,X,L. In more detail:-

KEYS "A", "P", "S". Pressing these keys prompts you to enter a number of a filename.

When ENTER is pressed, the attributes of that file will be set to either Archive, Read Only or System.

KEYS "E", "U". These keys are to Erase or Unerase a file. Enter the required File number and press ENTER and the file can be Erased or Unerased. If erased, a catalogue shows any erased files in INVERSE letters i.e. Blue letters on a WHITE background.

KEY "N". Pressing this gives displays the next page of files.

KEY "R". Press this key and you select a file number to rename - press ENTER then type in the new file name and press ENTER and the file will be Renamed.

KEY "D". Pressing "D" gives another catalogue.

KEY "F". This copies the format from one disc to another i.e. IT ERASES THE FILES - USE WITH CARE. Insert the original, press "F" when read at the prompt "INSERT BACKUP DISC" insert the new disc and press ENTER to format this disc.

KEY "X". This examines the current file in GREAT detail, listing all its attributes etc. It has a sub menu:- "Q"=QUIT; "L"=Load a file (saved by diskcat) from tape; "S"=Save the current file out to tape.

KEY "L". This Loads a file (previously Saved by diskcat) from tape to disc.

We hope you enjoy using "DISKCAT" and we are sure you will find it useful.

"SP5 TAPE TO +3 UTILITY 128K"

INSTRUCTIONS FOR USE

MEMORY MANAGEMENT IN THE SPECTRUM 128K

The Z80 processor in the Spectrum can only address 64K of memory at once. However, in the new 128K Spectrums, the computer contains 128K of RAM and 32K of ROM. The extra memory can be slotted in and out of the normal 64K at will, by a system called PAGING. This is achieved by setting the right bits in an Input/Output port - address 32765 decimal or 7FFDH was chosen for the new Spectrums. The memory map is shown below:-

DECIMAL	HEX
65535-----	FFFFH
RAM 0 - 7	
49152-----	C000H
RAM 2	
32768-----	8000H
RAM 5	
16384-----	4000H
RAM 0 - 1	
0-----	0

RAMs 2 and 5 are ALWAYS in the positions shown. The RAM banks are of two types; RAMS 4 to 7 are contended (i.e. they share time with the video circuitry) and RAM 0 to 3 are uncontended (i.e. the processor has exclusive use). Any machine code which has critical timing loops (such as music or communications programs) should keep all such routines in the uncontended banks. The bit field for the hardware switch at 32765 is:-

Bit No.	Function
0)
1) RAM Select
2)
3) SCREEN Select
4) ROM Select
5) 48K Lock

It is important to note that the paging in and out of RAMS and ROMS using this port CAN ONLY BE PERFORMED USING MACHINE CODE ROUTINES, AND CANNOT BE DONE FROM BASIC.

Bits 0,1 and 2 make a three bit number which selects which RAM goes into the C000H to FFFFH memory slot. In Basic, RAM 0 is normally in situ, and when editing, RAM 7 is used for various buffers and scratchpads. Bit 3 switches screens; screen 0 is held in RAM 5 (beginning at 4000H) and is the one that Basic uses, screen 1 is held in RAM 7 (beginning at C000H) and can only be used by machine code programs. Bit 4 determines whether ROM 0 (the editor ROM) or ROM 1 (the Basic ROM) is paged into 0 to 3FFFH. Bit 5 is a safety feature; once this bit has been set, the machine assumes a standard 48K Spectrum configuration and all the memory paging circuitry is locked out. It can only be returned to 128K by pressing RESET, or by switching off.

The main points to be aware of when transferring 128K programs are:-

- (1). ALWAYS consider where the Stack is - if you put it in RAM 7 then page it out, this will produce an immediate crash!
- (2). ALWAYS be aware which RAM and ROM you have paged or are going to page - check the appropriate bits.

ALL the programs on your SP5 disc will run on the +3 in +3 Basic.

NOTE:- Remember, to load any program:- At the +3 Menu, insert the SP5 disc and press ENTER - select the program to load by pressing the appropriate number.

We must thus look for the following or similar piece of code somewhere in the Basic Loader which will perform the memory switching i.e.:-

```
LD BC, 7FFDH - Loads BC register with I/O address.
LD A, 13H   - Load A register with data for switch - here it pages in Basic ROM
              and RAM 3.
OUT (C),A   - Perform the switch.
LD IX,C000H - Load IX register with Start Address of code to be loaded.
LD DE,4000H - Load DE register with length of code.
LD A,FFH    - Load A register with 255 i.e. code NOT Basic is to be loaded.
SCF         - Set carry flag - signal LOAD.
CALL 556H   - Call the ROM loading routine.
RET         - Return to Basic.
```

We will now illustrate the method by looking at:-

THE NEVER ENDING STORY - 128K.

Load the Header Reader as detailed earlier. From this we find:-

Tape Count	Filename	Type	Details
8 = 8	NEVER	Basic	T.L.=238, A.R Line 18
9 = 25	NEVTITLE	M/C	S.A.=32768; C.L.=6912
26 = 29	NEVLOADER	M/C	S.A.=42752; C.L.=512
38 ==)	Headerless Blocks		

The first task is to examine the Basic Loader "NEVER". All the initial operations are best performed in +3 Basic. Press RESET. Select +3 Basic, type load*t:, and load the file using MERGE**. This gave, on LISTING, as the more important points:-

```
10: CLEAR 24575: LOAD**CODE 16384
30: LOAD**CODE 42752
40: PRINT USR 42752
```

We see that Line 10 sets the stack at 24575, and loads the file "NEVTITLE" to the screen. Line 30 loads the file "NEVLOADER" to 42752, and line 40 starts the game. Thus "NEVLOADER" is the code to load and run the game. To examine this file, press RESET. Select +3 Basic, and type load*t:. Type CLEAR 24575, and load the file "NEVLOADER" by typing load** code. Load the KD1 disassembler, and examine the area starting at 42752 (A788 Hex). This shows the start of the machine code to load the rest of the game. The object now is to use this suitably modified code to load the rest of the code blocks and save them to disc, then write a suitable Basic Loader to reload these blocks from the disc. Looking at the code we see the following code blocks:-

RAM Paged in	Start Address	Code Length	Title
4	24576	18176	NEV1
4	43528	18496	NEV2
4	23552	888	NEV3
8	49152	16384	NEV4
1	49152	16384	NEV5
3	49152	16384	NEV6

Remember, IX holds the Start Address, DE holds the code length. At A787 we see a JP 6888H - the E.A. = 24576. We must now load each code block from tape using a modified "NEVLOADER" code, then save each one to disc. Return to Basic from KD1. Type CLEAR 42750 then NEW. This clears the KD1 Basic but leaves the "NEVLOADER" code intact. Select +3 Basic. Type load*t: then CLEAR 24575. We are now ready to load the code blocks from tape. Type in the following POKES:-

POKE	DETAILS
42753,0)	
42754,0)	- These 3 pokes remove the LD SP,5FFE, this avoids stack problems.
42755,0)	
42761,0)	- These 2 pokes remove the OUT instruction, which avoids
42762,0)	the paging in and out of any RAMS.
42776,251)	- These 2 pokes insert an EI (enable interrupts) and a RET
42777,281)	(return) to give a return to Basic after loading.

Type R.USR 42752 and PLAY in the block from tape. Type save"a:" and save to disc with save"NEV1"code 24576,18176. The names NEV1 etc are just convenient titles for each block. Type load*t:, then POKE 42778,243; POKE 42792,251; POKE 42793,281. Type R.USR 42778 and PLAY in the next block. Type save"a:" and save to disc with save"NEV2"code 43528,18496. We can ignore the block NEV3 since this merely overwrites the System Variables. Type load*t: then POKE 42817,243; POKE 42831,251; POKE 42832,281. Type R.USR 42817 and PLAY in the next block. Type save"a:" and save to disc with save"NEV4"code 49152,16384. Type load*t: then POKE 42826,2 and R.USR 42817 and PLAY in the next block. Type save"a:" and save to disc with save"NEV5"code 49152,16384. Finally, type load*t: then POKE 42826,3; R.USR 42817 and PLAY in the last block. Type save"a:" and save to disc with save"NEV6"code 49152,16384. We now have the required blocks of code on disc, it only remains to write a suitable Basic Loader to reload them. Care must be taken with the order of reloading of the blocks, and the positioning of the stack. The following Basic Loader achieves both these points. Press RESET, select +3 Basic and type in:-

```
10: CLEAR VAL*43510"
20: LOAD"NEVSCR"CODE VAL*16384",VAL*6912"
30: GOSUB VAL*200"
40: LET A=VAL*8":GOSUB VAL*400"
50: LET A=VAL*1":GOSUB VAL*410"
60: LET A=VAL*3":GOSUB VAL*420"
80: POKE VAL*42765",VAL*170":POKE VAL*42768",VAL*41":LET A=VAL*4":GOSUB 430
90: RESTORE 310:GOSUB 200
95: LOAD"NEV1"CODE VAL*24576"
100: PAUSE VAL*100":RANDOMIZE USR VAL*42752"
200: LET A=VAL*42752"
210: READ A$:IF A$="*" THEN RETURN
```

```

220: POKE A,VAL A$:LET A=A+1:GOTO 210
300: DATA "243","62","1","1","253","127","237","121","33","144","101","17","0",
"192","1","0","64","237","176","62","16","1","253","127","237","121",
"251","281","*"
310: DATA "243","62","4","1","253","127","237","121","49","254","95","195","0",
"96","*"
400: POKE VAL"42754",A:LOAD"NEV4"CODE VAL"26000":RANDOMIZE USR VAL"42752":RETURN
410: POKE VAL"42754",A:LOAD"NEV5"CODE VAL"26000":RANDOMIZE USR VAL"42752":RETURN
420: POKE VAL"42754",A:LOAD"NEV6"CODE VAL"26000":RANDOMIZE USR VAL"42752":RETURN
430: POKE VAL"42754",A:LOAD"NEV2"CODE VAL"26000":RANDOMIZE USR VAL"42752":RETURN

```

Save this to disc to auto run from line 10 with save"NES"line 10. The program will reload with load"NES".

Points to Note:-

- (1). Line 10 sets the stack to 43518 which is a free area and allows plenty of room for the Basic Loader.
- (2). Line 20 loads the screen picture.
- (3). Line 30 POKES the following routine to address 42752:-

```

DI          - Disable interrupts
LD A,1     - Load A with 1;changed by POKES
LD BC,32765 - Load BC with I/O address
OUT (C),A  - Page in correct RAM = 1 here
LD HL,26000 - Load HL=26000,start of code
LD DE,49152 - Load DE=49152,move code here
LD BC,16384 - Load BC=16384,length to move
LDIR      - Move code from 26000 to 49152
LD A,16    - Load A with 16
LD BC,32765 - Load BC with I/O address
OUT (C),A  - Page in RAM 0 (Basic RAM)
EI          - Enable interrupts
RET        - Return to Basic

```

- (4). Lines 40,50 and 60 POKE the correct value into 42754 (the value to go into the A register). The routine at lines 400,410,420 and 430 loads the block of code from tape, calls the machine code to page in the correct RAM, moves the code into place using the LDIR, pages back in the Basic RAM 0, and returns to Basic.
- (5). Line 80 alters the routine to move "NEV2" code of length 10496 to 43520.
- (6). Line 90 pokes the following routine into address 42752:-

```

DI          - Disable interrupts
LD A,4     - Load A with 4; page in RAM 4
LD BC,32765 - Load BC with I/O address6
OUT (C),A  - Page in RAM 4
LD SP,24574 - Set stack to 24574
JP 24576   - Jump to start of game

```

- (7). Line 95 loads the code "NEV1" into place. Line 100 pauses, then jumps to the above routine and starts the game.

Using similar methods and considerations, other 128K games may be transferred to your +3 Drive.

"KOBRAHSOFT KD1 DISASSEMBLER"**INSTRUCTIONS FOR USE****Introduction.**

The KD1 disassembler is an efficient and easy-to-use disassembler for your ZX Spectrum. It can be used to list machine code instructions on your T.V. Screen, or output them to a ZX Printer. It allows easy interchange to and from Basic, and also contains a Number Converter to convert (a) Hexadecimal to Decimal numbers, (b) Decimal to Hexadecimal numbers.

Loading.

To load KD1, first ensure the Spectrum is reset, by pressing the RESET switch on the left side.

Insert the SP5 disc, then press ENTER. The SP5 Menu program will load and run automatically.

At the SP5 Menu, to load the KD1 disassembler, press key "9" then press ENTER - KD1 will load and run automatically.

When KD1 has loaded the following message will appear:- "To Start: Press BREAK". On pressing BREAK, KD1 goes into the Command Mode - with a flashing cursor at the bottom of the screen. At this stage, the following commands are available:-

Commands.**(1). Number Converter. (N).**

To access the Number Converter, press "N". The message:- HEX or DEC? is displayed. To convert a Hexadecimal to a Decimal number, press "H". Now, enter your Hex. number. NOTE:- Any leading zeros MUST be included i.e. enter 38 Hex. as 0038, etc. Any normal Hex. letters (A-F) may also be entered. On pressing ENTER, the result is shown. To convert Decimal to Hex., press "N", then "D" for Deciaal. Now, enter your Decimal number WITHOUT leading zeros. Typing ENTER gives the result.

(2). Disassembler. (D).

NOTE:- Any section of memory may be disassembled, but NOT that occupied by KD1. Any attempt to do so will produce the message:- INVALID ADDRESS. To enter the disassembler mode, press "D". Next, enter, IN 4 HEX. DIGITS, your required start address. You can now also specify similarly a 4 HEX. DIGIT end address. If you don't, the only difference is that KD1 will continue its disassembly up to 65535 (FFFFH), which is probably inconvenient especially when outputting to your printer! In the case of no specified end address - press ENTER. You are then asked if you require a printout to the ZX Printer. If yes - press "\", if no - press N, when the listing will appear on the screen. To list more - press ENTER. To end the listing - press F.

When an end address is specified - ENTER is not needed - the enquiry PRINT? appears on pressing the last digit. NOTE:- To get "\": press Symbol Shift and "V" keys.

If you do NOT specify an end address, you can stop a continual printout to a ZX Printer by pressing "BREAK". This will return you to Basic, with the usual error message.

Return to Basic. (R).

For a NORMAL return to Basic, press "R" while in Command Mode - you will be returned to Basic. To re-enter KD1 from Basic, type:- RANDOMISE UBR 59625, then press "BREAK". This will take you into the Command Mode.

"HEADER READER"**INSTRUCTIONS FOR USE.****LOADING:-**

Proceed as for KD1 above, but at the SP5 Menu, press key "8" then ENTER - the Header Reader program will load and run.

USING:-

The HEADER READER will read the data from the header section at the start of each data block in a program. It will display details such as:-

FILENAME:- The program name. This may sometimes be printed vertically due to the presence of certain control codes in the header e.g. CHR\$(13), etc.

PROGRAM TYPE:- i.e. Basic, Machine Code, SCREEN\$, Numeric Array, Character Array, etc.

PROGRAM LENGTH:- The HEADER READER will give, for a BASIC program, the total program length (Basic program length + Variables), and the normal program length. It will also give the length of a machine code block.

START ADDRESS:- For a machine code block, this is the start of the block in memory.

AUTO-RUN LINE NUMBER:- For Basic programs only.

To obtain this information, load the HEADER READER as described above, then load your desired cassette and press "PLAY". For each header, the screen will clear, and the data read will be displayed.

It is usually best to "STOP" the tape when each header is read, so that the data can be written down. Press "PLAY" to continue. Repeat until no more data loads in i.e. the program has finished. This can then be repeated with any other tape which you wish to investigate.

We recommend you use the HEADER READER before copying a program, since this will tell you how many data blocks you must copy and hence, when the program has ended.

NOTE:- DO NOT, AGAIN, PRESS "BREAK" AT ANY TIME!

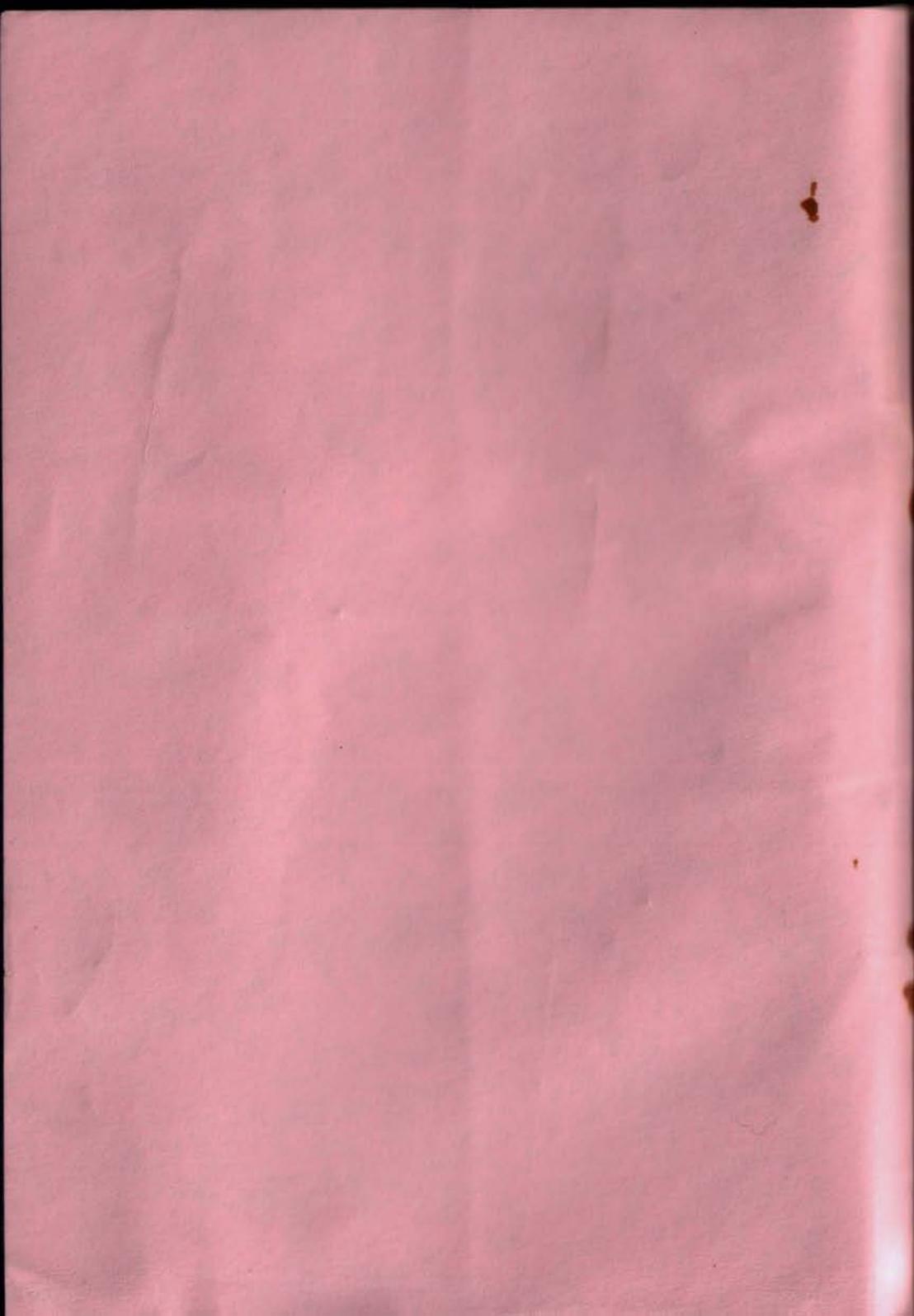
"HEADERLESS BLOCK LENGTH READER"

INSTRUCTIONS FOR USE.

TO LOAD:-

Proceed as for KDI earlier but at the SP5 Menu press key "7" then ENTER. The HBReader program will load and run displaying the message "PLAY the Tape". In order to determine the length of a Headerless Block, position your tape at the start of the block, then press "PLAY" on your recorder. The program will read in the bytes, count them, and print out the number of bytes in the block. To read in another block, press "r", then repeat as above.

With these two utilities, the composition of most programs can be determined (except fast loaders and pulsing programs).



THE HISTORY OF THE
REPUBLIC OF THE UNITED STATES
OF AMERICA
FROM 1776 TO 1876
BY
JAMES M. SMITH
VOLUME I
1876