

# ⑧ MICRODRIVE EXCHANGE

G.A. BOEKER. 29 Chadderton Drive, Unsworth, Bury, Lancs. Tel: 061-766 5712

The following is a shortened version of the one on MAGIC-MESSAGE cassette and will be referred to several time in this issue.

## FALSE HEADER MAKER

This short program will make a False Header for Basic or Machine Code programs. It will call the program "NEW HEADER"

```
10 RESTORE: DATA C,78,69,87,32,72,69,
    65,68,69,82,A,B,0,128,0,0,0,221,33,
    48,117,17,17,0,175,55,205,194,4,201
20 INPUT "0=BASIC 3=M/CODE ";C
30 IF C<>3 AND C<>0 THEN GO TO 20
40 INPUT "ENTER LENGTH ";L
50 PRINT AT 10,7;("BASIC" AND NOT C)+
    ("M/CODE" AND C);"...Length=";L
60 LET B=INT (L/256): LET A=L-B*256
70 FOR J=30000 TO 30030: READ X:
    POKE J,X: NEXT J
80 PRINT "" SET TO RECORD, THEN PRESS ENTER"
90 FOR J=1 TO 150: NEXT J
100 PAUSE 0: RANDOMIZE USR 30018
```

Simply enter this program, RUN and obey the video.

This message means place Blank tape in recorder, set to record, then press Enter key.

## ANT ATTACK TO MICRODRIVE (Headerless-File)

This demonstrates the technique which can be used on most super-long programs to chop them into 3 parts. By Loading program 4000 Bytes higher we lose the top 4000 Bytes. Save 1st 4000 and middle part. Then Load in so last 4000 go onto screen, then SAVE them. The main Code section starts at 23552 and is 41984 Bytes long.

1. Make up a "False" header using the FALSE HEADER MAKER program for Machine-Code of 41984 Bytes. Enter RUN USR 0
2. Load False header by; CLEAR 27551: LOAD "" CODE 27552
3. Wind ANT tape to be at start of the long Headerless-File, then play in all the File.
4. When loaded, save by; SAVE\*"m";1;"ANT1" CODE 27552,4000 Then Save; SAVE\*"m";1;"ANT2" CODE 31552,33984

This has Saved all except top 4000 Bytes. Rewind ANT tape to be at start of the File again.

5. Load False Header by; CLEAR 43936: LOAD "" CODE 43937 then play in the File from the tape again. The last 4000 Bytes will eventually appear on the screen. This is part we want.
6. Save by; SAVE\*"m";1;"ANT3" CODE 16384,4000

We now have first 4000 Bytes in ANT1, middle section in ANT2 and last 4000 Bytes in ANT3.

7. Type in this basic loader;

```
10 CLEAR 27551: LOAD*"m";1;"ANT2" CODE 27552
20 LOAD*"m";1;"ANT3" CODE 61535
30 LOAD*"m";1;"ANT1" CODE 16384
40 DATA 17,0,92,33,0,64,1,160,15,237,176,195,8,151
50 FOR J=23300 TO 23313: READ X: POKE J,X: NEXT J
60 LET X=USR 23300
```

NOTE:  
The order  
of Loading  
is correct.

Save to microdrive by; SAVE\*"m";1;"ANT" LINE 10

NOTE: The 195,8,151 at end of the DATA statement causes a JUMP to location 38664 which starts the game.

If you are the inquisitive type, then change the 195 in the DATA statement to be 201. Try it and see what happens.

## TORNADO LOW LEVEL TO MICRODRIVE

This Starts at location 16384 & is 49052 Bytes long and also it is a Headerless-File. It is far too long to be chopped-up by the methods previously used. (Obviously a full 48K program would give same problems). K.R.WALKER of Northfield, Birmingham solved this problem by putting a Loader into the few remaining Bytes of the memory and instead of returning to Basic, causes a Jump to ERROR routine, and this causes program to STOP.

1. Type in the following;
 

```
10 CLEAR 65535
20 DATA 221,33,0,64,17,156,191,62,255,55,205,86,5,207
30 FOR J=65430 TO 65443: READ X: POKE J,X: NEXT J
40 RANDOMIZE USR 65430
```
2. Position your TLL tape to be just after the end of the Basic then RUN the above program and play in TLL File. When all in, an error message will appear at bottom of video.
3. TLL main code in program actually starts at 26300. Save it by  
SAVE\*"m";1;"TLLc" CODE 26300,39136
4. Type in this Basic Loader;
 

```
10 CLEAR 65535
20 BORDER 0: PAPER 0: CLS
30 LOAD*"m";1;"TLLc" CODE 26300
40 RANDOMIZE USR 35354
```

Save this Loader by; SAVE*"m";1;"TLL" LINE 10
--

NOTE: It is the 207 which causes Spectrum to do an RST 8 which causes program to STOP. This method may not work on all programs

#### RESTORE n & WAR OF THE WORLDS TO MICRODRIVE

Whilst this is a badly written game with jerky graphics, method used to transfer demonstrates use of RESTORE n to read required DATA. Main part of this game is the Bytes sections called WOW MC Games main parts are transferred by;

```
LOAD "WOW MC" CODE 30000: SAVE*"m";1;"WWc" CODE 30000,21536
```

Rewind tape and MERGE in the Basic part by using: MERGE "w1"

Altho' a "modern" game, it will NOT work if an interface 1 is fitted, so we need to reclaim the microdrive map area. This is done by putting a routine in a line 9999 thus:

```
9999 RESTORE 9999: DATA 33,240,92,17,182,92,195,229,25:
```

```
FOR J=23296 TO 23304: READ A: POKE J,A: NEXT J:
```

```
RANDOMIZE USR 23296: RESTORE: GO TO 1
```

Save to microdrive by; SAVE\*"m";1;"WWb" LINE 9999

Note that the RESTOREs in line 9999 ensure that we Read the Data from correct line. The RESTORE at the end ensure that the main program knows to start Reading Data from its usual position.

Finally make up & save a Loader for the game as;

```
10 CLEAR 43999: LOAD*"m";1;"WWc" CODE 44000: LOAD*"m";1;"WWb"
```

#### MICRODRIVE-V-WAFADRIIVE

The Wafadrive is well constructed in a box about 4" X 4" X 8" & contains TWO drives plus an RS232 & Centronic port AND includes in its ROM the routines for doing LPRINT & LLIST to the ports. Unit connects to Spectrum via a short ribbon cable. (Why didn't Clive think of that)? It is claimed to be more reliable than the Microdrive and appears better value for money in that you get two drives plus better port facilities. It is slower than a M/drive if you simply plonk a multipart game onto it, mainly due to the system Loading in a Directory first (similar to a CAT).

(The NEW 007WD tape has been designed to speed-up this Loading)

The Wafadrive CANNOT Save a program of one Byte! Easiest way around this problem is to first Load any one Byte program at 30000. Enter PRINT PEEK 30000 then in your Basic Loader replace the LOAD\*"name", etc, by POKE x,y where x is the location the 1 Byte is to Load into and y is the value we found.

Using ATIC-ATTACK which has five sections, I timed the Loading from tape, Microdrive and Wafadrive. (The Wafadrive time was from the instant the colours appeared on screen). Wafa used was a 64K; 128K Wafa would take longer to Search for Directory. Four timing checks as follows:

1. Normal time to load ATIC ATTACK from a tape.....4mins 21secs
  2. Time approx to Load from a Microdrive.....0mins 20secs
  3. Time approx if simply plonked on a Wafa.....2mins 02secs
  4. Program transferred to Wafa using 007WD. Loads in just 25secs
- SUMMARY:..Which ever system you have, don't be jealous of other.

MICRODRIVE-EXCHANGE was started as a Microdrive club and will remain so. The techniques printed do apply to the Wafadrive and occasional BRIEF comments on such will appear.

#### MACHINE-CODE EXPLAINED.....BLOCK MOVER

The most used routine is the Block Mover used to Move programs which had been temporarily relocated whilst Microdrive was being used. By machine-code we could move a full 64K in less than one second (don't know where we'd move it to)!

The machine-code program appears as a string of numbers in consecutive locations. If the first location was 30000, then a RANDOMIZE USR 30000 would cause Spectrum to start obeying what ever is in 30000 onwards as machine-code. For convenience, the numbers are put into a DATA statement and POKEd into memory. We will move 6912 Bytes starting at 40000 down to 16384. Program is

```

10 DATA 17,0,64,33,64,156,1,0,27,237,176,201  ←The numbers.
20 FOR J=30000 TO 30011:READ X:POKE J,X      ←POKEs to memory
30 RANDOMIZE USR 30000                        ←Starts action.
```

The Spectrum sees the numbers as instructions as shown below:

NUMBERS	NEMONICS	DESCRIPTION.
17,0,64	LD DE,16384	Load DE register pair with 16384
33,64,156	LD HL,40000	Load HL register pair with 40000
1,0,27	LD BC,6912	Load BC register pair with 6912
237,176	LDIR	Load, increment then repeat.
201	RET	Return, in this case, to Basic

The DE pair are set to DEstination.....16384

The HL pair are set to FROM location.....40000

The BC pair are set to number of Bytes to be moved...6912

LDIR is a very powerful instruction. It puts the value it finds at location pointed to by the HL pair into the location pointed to by the DE pair. It then increments (means it adds 1) to both HL & DE so that they point to next address, then it decrements BC and if BC not equal to zero, action is repeated. When BC=0 the next instruction is obeyed. Note that the LDIR is NOT really on its own but depends upon the DE, HL & BC registers being set-up to the required values first.

How do I know these numbers are these Codes? Possibly because I'm brilliant (and only held back by my extreme modesty) Or possibly I looked them up in a list of Z80 instructions. Such lists appear in any machine-code book, but you can find them in the Character set list in your Spectrum manual. This starts at page 183 of older books, or page 135 in newer books. Obviously has to be read carefully. 17 has under the column Z80 ASSEMBLER ld de,NN N stands for a Number. NN stands for a double number. Therefore 17 on its own is NOT complete as it needs TWO numbers after it. In our example we have, 17 0 64 and the value of the two numbers are First number + 256\*Second number. The is 16384. Checking the list for numbers 33 & 1 will show our coding is correct. Checking 237 shows nothing as there are too many Z80 codes beginning with 237 (Hex ED) to list them. If this occurs, check the next number. This is 176 in our example, but remember it was after 237 (ED in Hex). The listing for 176 under the last column marked; "After ED" states that it is LDIR.

Just to explain another Z80 method, look up 42 in the list and we find ld hl,(NN) the brackets means the CONTENTS of location pointed to by NN. 33,64,156 was load HL with 40000 as a value. 42,64,156 would be load HL with the CONTENTS of whatever is in location 40000.....Equivalent to a Load HL, PEEK 40000.

#### DEMONSTRATION OF SPEED OF THE MOVER PROGRAM.

Type in this program:

```

10 LOAD*"m";1;"SCREEN" CODE 40000
20 RESTORE: DATA 17,0,64,33,64,156,1,0,27,      Save to m/d:-
    201: FOR J=30000 TO 30011:                     SAVE*"m";1;"DEMO"
    READ X: POKE J,X: NEXT J
30 PAUSE 0: RANDOMIZE USR 300000
```



Take a game tape with a SCREEN\$ part and Load this Screen\$ by;  
 LOAD "" CODE 40000: SAVE\*"m";1;"SCREEN" CODE 40000,6912

When program DEMO is Loaded and ran, the SCREEN program Loads in to 40000 then program halts. Press Enter key & picture will be Block Moved from 40000 to the screen area instantly.

Duncan Sarens of Kensal Rise, London, makes his Basic Loaders auto-run but makes first line; IF INKEY\$="s" THEN STOP  
 Load the program by usual command, then press and hold s key as soon as Microdrive starts. Program will Load and then Stop.

S.WATTS of Wickford, Essex, solved the problem of games such as PINBALL which should, but don't autostart after a Block move. Instead of using RANDOMIZE USR xxxxx for the Block Mover, use instead LET X=USR xxxxx. This method worked also with FRED, ASTRO BLASTER, G-FORCE, ARMAGEDON to name just a few. Using RANDOMIZE USR or LET X=USR should give same action, but as the LET X=USR works on more it would be better to always use in Basic Loader.

E.J.F.AUSTIN of Stafford, FORMATS his cartridges a few times, and then does a CAT to find number of K. Finally he re-FORMATS with the title required with last two characters being number of K.  
 VU-FILE PROBLEMS/SOLUTION. From info sent in by members I have been able to prove nothing! I suspect that there are versions of this which have same lengths as later versions but different in part of Coding. To avoid piracy, if your VU-FILE is giving the problem with option 6, send me your original tape and I will put on the version I have which appears to work correctly. The cost for this service will be an 18p stamp. Send tape in a padded bag.  
 (Previous members using my VU-FILE version say all works O.K.)

4000 BYTES ON SCREEN. Instead of putting 3000 Bytes temporarily onto Screen in chopping-up long programs, in future please use 4000. This will ensure routine will work on a WAFADRIIVE as well.

MICRODRIVE or WAFADRIIVE tip. After typing a long line with LOAD commands, etc, the Spectrum refuses to let you enter it as you are not in Microdrive or Wafadriive mode. To avoid having to rub-out line, then enter CLS# or NEW \* for Wafadriive, and then re-entering this long line. Move cursor to be just after the line number, press REM then enter the line. Enter the CLS# or NEW \*, edit down you line and delete the REM from it.

DON'T EVER TRY TO SAVE TIME BY USING FORMAT IN A BASIC LINE TO FORMAT SEVERAL CARTIDGES. If you forget to delete such a line the next cartridge you try to load by "RUN" will be wiped clean. This accident has happened to several members recently.

K.C.RYLETT of Burnage, Manchester, sent in a Screen Dump for Dot-Matrix printer GP-80A (7\*5) Parallel printer with Kemson inter/f Most likely wont run on other printers without modification.....  
 ..sorry, I cannot help on this matter. Two programs; first gives straight dump and second rotates picture thro 90 degrees. Takes between 17 to 30 minutes to print out. Program is too long to print in MDX. Send two 13p stamps for a photo-copy.

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

ALL OF MICRODRIVE-EXCHANGE IS COPYRIGHT & MAY NOT BE REPRODUCED WITHOUT WRITTEN CONSENT OF THE EDITOR. (WILL NOT BE GIVEN).