

Editor: G.A.BOBKER of ZX-GUARANTEED

-----

This the last issue of Microdrive-Exchange as it just would not be realistic to attempt to continue such a Newsletter. The Blame for this has to be placed on the Hardware Units, (including the Interface 007), which have resulted in people simply transferring programs to Microdrive without having to derive manual methods to do so. The "protection" in modern software gives no other choice. Even using the 007-SPY DEPULSER 1 or 2 tape did NOT always make transfers possible, and it would be very unfair to have all methods based on having to use just one of my own products (would be nice for me)! "Transfer Tapes" such as Medsoft, Trans-Express and Lerm products all have limitations, as does the 007MD (but this was admitted). Medsoft has long since disappeared but suprisingly the Lerm is still advertised. ZX-GUARANTEED has never had any serious competition. Part time producers of tapes, such as Lerm, just do not put in sufficient time to make their items "user friendly" (despite the fact that Lerm included a 40 page booklet, which would have been easier to understand if written in Chinese).

-----

#### AWARDS BASED ON MEMBERS LETTERS RECEIVED

MOST AWKWARD INTERFACE: SPECMATE. It turned even a 1 Byte prog into a full 48K. (Sheer ignorance or just bad programming)?

MOST BAFFLING PROGRAM: "FRIENDLY FACE". (This was sold a few years ago). Its main claim to fame was that it could Transfer TASWORD to Microdrive! Big deal. Even the original Tasword jumped onto Cartridge easily with only the LOAD/SAVE/VERIFY commands being changed. Couldn't find any other use for it!

SHORTEST YET MOST POWERFUL BASIC: "SEARCH" which is a Pseudo Disassembler which found the Loaders for Headerless-Files and printed them out in plain English. Written by ZX-GUARANTEED (Big 'ead), Which proves that I am only held back by my modesty).

MOST UNFRIENDLY PRODUCTS: No prizes for naming company which over the years appears to have deliberately set out to confuse the Spectrum Users.....one clue. Its a four letter word.

-----

#### JACK THE NIPPER TRANSFER by S.J.NUTTING of HISTON,CAMBRIDGE

The program contains two parts. First is a 372 Byte Turbo Loader in the Variables Area, and is Block Moved to 65293 onwards. The values in the Basic are made to look wrong just to be awkward. Main Headerless part is turbo loaded into 16384 to 65535 which means it overwrites the Loader. (Minor details like these don't seem to affect Mr.S.J.N)!

1. Type in the following;

```
10 CLEAR 64999: FOR J=65000 TO 65015: READ A: POKE J,A: NEXT J
20 DATA 55,62,255,221,33,199,254,17,116,1,205,86,5,48,241,201
30 RUN USR 65000
40 CLEAR 65249: POKE 65294,0: POKE 65295,0: POKE 65296,0
50 POKE 65302,126: POKE 65303,190: POKE 65309,0: POKE 65310,254
60 FOR J=65250 TO 65258: READ A: POKE J,A: NEXT J
70 DATA 33,167,97,34,178,92,195,183,17
80 RUN USR 65293
```

2. Set your JACK tape to be just AFTER the very first Header on the tape. RUN the above then play in all the rest of the JACK tape. When all Loaded in, Spectrum will NEW but the CODE is still safely held above RAMTOP, which the machine-code has set to be 24999 (by line 70 action).

3. Save the code by entering; SAVE\*"m";1;"JACKc" CODE 25000,40150

4. Type in this Basic Loader;

```
10 CLEAR 24999: LOAD*"m";1;"JACKc" CODE 25000: RUN USR 25000
5. Save this Loader to Microdrive by; SAVE*"m";1;"JACK" LINE 10
```

Note the RUN USR. Generally only used if NOT returning to Basic. S.J.N. used it, and it works! Line 20 is to Load the Basic Bytes into 65000 onwards, moving the Code in it to correct address. The DATA in Line 70 is effectively a CLEAR 24999: NEW. This is;

LD HL,24999	This sets 23730 & 23731, the System Variable
LD (23730),HL	holding RAMTOP, to be 24999. The CALL 4535 is
CALL 4535	a call to the NEW routine in the ROM.

#### BOUNDER TRANSFER

As this game is a Turbo Loader and a full 48K, Original loader is modified to load in the game part lower into the ROM (into addr 9472) so that effectively Screen part is chopped-off, then waits for you to press a key, then Saves two Files to a tape.

1. Type in this program;

```
10 LOAD "" CODE 61074
20 DATA 243,221,33,0,37,17,0,192,62,243,55,205,184,238,243
30 DATA 62,191,219,254,31,56,249 <----Waits for Enter Key
40 DATA 221,33,0,80,17,0,149,62,255,205,194,4,243
50 DATA 33,255,255,43,124,181,32,251 <----Gives a delay.
60 DATA 33,255,255,43,124,181,32,251 <----doubles the delay.
70 DATA 221,33,0,64,17,0,16,62,255,205,194,4,195,0,0
80 FOR J=61435 TO 61500: READ A: POKE J,A: NEXT J
90 RANDOMIZE USR 61435
```

2. RUN the above then play in all the BOUNDER tape from the very beginning (the Basic will be ignored).

3. Place a blank tape in recorder, set to record, then press the Enter key and two Headerless-Files will be Saved out with a gap of approx 1 seconds between then. First File will be 38144 Bytes long, and second will be 4096 Bytes. When both have been Saved out, the Spectrum will NEW itself.

4. Type in the following;

```
10 CLEAR 27391
20 DATA 221,33,0,107,17,0,149,62,255,55,205,86,5,201
30 FOR J=23300 TO 23313: READ A: POKE J,A: NEXT J
40 RANDOMIZE USR 23300
50 SAVE*"m";1;"BOUNDER1" CODE 27392,38144
60 RANDOMIZE USR 23300
70 SAVE*"m";1;"BOUNDER2" CODE 27392,4096
```

5. RUN the above then play in the Headerless File tape just made When Microdrive starts up, STOP the tape. After first File has been transferred, play in the second File.

6. Type in this Basic Loader;

```
10 CLEAR 27391
20 LOAD*"m";1;"BOUNDER1" CODE 27392
30 LOAD*"m";1;"BOUNDER2" CODE 16384
40 DATA 49,255,95,33,0,64,17,0,91,1,0,16,237,176,195,40,135
50 FOR J=20480 TO 20496: READ A: POKE J,A: NEXT J
60 RANDOMIZE USR 20480
```

Save to Microdrive by; SAVE\*"m";1;"BOUNDER" LINE 10

Long programs such as this, give the problem that the Stack is overwritten. The method used here is to set SP to be 24575 which was found in the turbo Loader. Finding the "start" address is usually the most awkward part. My method is to look through the program and try to find the part waiting for keys to be pressed, and assume it might be the menu part. (Look for CP instructions and/or BIT tests). Next alternative is to get the Code part from around 23755 and turn it into Basic, and hopefully a RUN USR is contained in this Basic. In this program as an example, the BOUNDER1 is temporarily Loaded onto the screen then the Block

Mover in the Data statement moves it to its correct location of 23296. Loading it directly into correct location would lock-up the Spectrum. Instead, Load it 10,000 Bytes HIGHER by entering; LOAD\*"m";1;"BOUNDER1" CODE 33296. The Basic, if there is any, is now sitting at 33755 if game was written without an Interface 1 fitted, or 33813 if was fitted. The System Variable 23635 and 6 points to the start of the Basic. By Poking them to point to new value, we can "reveal" the Basic. To look at it here, we use;

```
10 LOAD*"m";1;"BOUNDER1" CODE 33296
20 POKE 23636, INT (33755/256): POKE 23635, 33755-PEEK 23636*256
```

RUN this (after you've transferred BOUNDER), and a Basic is then revealed, which showed a CLEAR 24575. Also note that "BOUNDER1" block includes the original System Variables. This means that we can also now check the RAMTOP values which are now in locations 33730 and 33731 (by entering PRINT PEEK 33730+256\*PEEK 33731). A RANDOMIZE USR was found in the revealed Basic, but is the one used by the writers to Save out the full 48K. Using a Disassembler on the area around this Randomize value showed a JP 34600, which by experiment was discovered to be the "start" address for Bounder. Sounds easy! Actually I wrote down 15 addresses I thought could be the value to start the game. Luckily the 34600 was first. NOTE: This method of turning the Bytes into Basic is not that reliable as altho' you get a chance to look at the Basic, if you mess about with it too much, it all dissappears or corrupts.

#### NIGHTMARE RALLY TRANSFER

The latest Speedlok programs can ONLY be transferred after they have been "depulsed" by the new DEPULSER 2 program. These three programs will do them all with the only change being the values used in Line 50 of PROG3. See the end note about this.

<pre>10 DATA 243,221,33,0,64,17,0,165 62,255,55,205,86,5,243,62,191, 219,254,31,56,249,221,33,0,80, 17,0,149,62,255,205,194,4,243, 33,255,255,43,124,181,32,251, 221,33,0,64,17,0,16,62,255,205, 194,4,195,0,0 20 FOR J=60000 TO 60057: READ A: POKE J,A: NEXT J: RUN USR 60000</pre>	<pre>PROG1</pre>	<ol style="list-style-type: none"> <li>1. Load and RUN "PROG1".</li> <li>2. Play in ONLY the last long Headerless-File on the Back-Up tape.</li> <li>3. Place a blank tape in tape recorder and set to record, then press the Enter key. Two Files will be Saved out. When completed, Spectrum NEWS</li> <li>4. Load and RUN "PROG2".</li> <li>5. Play in the Files you have just Saved. STOP the tape when M-drive starts running. Play in second File when microdrive stops.</li> <li>6. Load in "PROG3" and save it to m-drive by; SAVE*"m";1;"name" LINE 10</li> </ol>
<pre>10 DATA 221,33,144,101,17,0,149, 62,255,55,205,86,5,201 20 FOR J=23300 TO 23313: READ A: POKE J,A: NEXT J 30 RANDOMIZE USR 23300 40 SAVE*"m";1;"name1" CODE 26000,38144 50 RANDOMIZE USR 23300 60 SAVE*"m";1;"name2" CODE 26000,4096 70 RUN USR 0</pre>	<pre>PROG2</pre>	
<pre>10 CLEAR 27391 20 LOAD*"m";1;"name1" CODE 27392 30 LOAD*"m";1;"name2" CODE 16384 40 DATA 33,0,64,17,0,91,1,0,16, 237,176,195 50 DATA 106,224 60 FOR J=20480 TO 20493: READ A: POKE J,A: NEXT J 70 RANDOMIZE USR 20480</pre>	<pre>PROG3</pre>	<p>I intend making a list of the values for Line 50 for various games. Members may write or telephone for them. Please let me know any values you find &amp; they will be passed-on to callers.</p>



The DATA line 50 is correct for NIGHTMARE RALLY. Values for most of the new type games should be available from end of October.

### ULTIMATE BORDER EFFECTS

At last I've cracked the ultimate Border Loading effects just in time for this last issue. After trying this you wont want any other colours.

1. Load in your CHECKER program (MERGE it in if you made it auto start).
2. Change lines 20 and 30 to be;  
20 FOR J=51540 TO 51533 STEP-1: POKE J+2,PEEK J: NEXT J  
30 POKE 51533,237: POKE 51534,95
3. Erase old CHECKER and save this modified version as CHECKER.

What the above does is to simply add in a LD A,R before the Border is sent out, giving what I consider to be the Ultimate Border effect. Try it by RUNNING this new CHECKER and then Loading a Screen (without its Header).

If you still want to experiment with Borders, after trying out the modifications above, then use original CHECKER and the POKES below, (Sent in by S. M. GOODMAN of LEEDS). The following POKE addresses in the HIGHROM control the Leader stripes and Loading stripes. Experiment using different values. The original values are shown in the brackets.

51384 (2)      51441 (3)      51534 (8)      51536 (8).

### OPUS CONVERSIONS

A fair number of members have switched over to the OPUS now it has dropped in price. Note that most of the routines in MDX will work on the Opus without modifications, but quite a few include the "reclaim" routine. Simply omitting this is all that is necessary to make routine suitable for the OPUS. If any problems with this, then please note that I will shortly have available a free leaflet listing the necessary changes. (Please send a stamp and request MDX OPUS LEAFLET). The RANDOMIZE USR 4007 built into the OPUS should switch in 58 extra Bytes to make Microdrive type programs work without problems, but it does NOT always work. Possibly a CLEAR within the programs reset it?

To B or not to b that is the question often asked. It does NOT matter. The Spectrum does NOT know the difference between large or small variable names. 10 LET A=10: PRINT A: PRINT a on the screen will be printed 10 in each case.

\*\*\*\*\*  
\* I will be starting a new club dedicated to the learning of \*  
\* Z80 MACHINE-CODE. Each member would initially receive a tape \*  
\* containing a Dissassembler, a standard Loader and possibly a \*  
\* few exercises. Membership in U.K. would be £15. (This is for \*  
\* the FULL course). MDX members joining before end of 1986 are \*  
\* offered a £1 membership discount. Expected to start in early \*  
\* November 1986. Z80 routines would be based on the Spectrum & \*  
\* Amstrad. The method used will be in my usual unconventional \*  
\* style by being done not in the usual logical (boring) order, \*  
\* but instead will be presented in the more interesting order. \*  
\*\*\*\*\*

ALL OF MICRODRIVE-EXCHANGE IS COPYRIGHT AND MUST NOT BE COPIED  
OR REPRODUCED WITHOUT PRIOR CONSENT OF THE EDITOR (NEVER GIVEN)