

Editor: G.A.BOBKER of ZX-GUARANTEED

Amstrad is launching TWO computer in September 1986. On the 12th will be an IBM PC Compatible complete with monitor & Disc drive. At end of September the revamped Spectrum 128K is due. The latter should revitalise the ailing Spectrum market. The PC compatible at £399 should knock out ALL other IBM PC clones, and there is now a possibility that all future home computers will be PC compatibles? For business purposes there is a vast range of fairly cheap software for such machines and it makes sense to get a home computer with the possibility to convert to business use at a later date, AND it has 512K of memory. Albeit that the Amstrad range of home computers use the Amstrad obnoxious basic and they certainly DON'T deserve to be successful. I regret that they WILL do exceptionally well with these machines. Come back Uncle Clive, all is forgiven!

IBM, in order to ensure that they set the computer industry standard, freely encouraged other companies to make compatible computers. They did. Now IBM have announced that they are to get out of the small computer market as they cannot compete!

MAX HEADROOM TRANSFER by S.J.NUTTING of HISTON,CAMBRIDGE

Program is in the block called "g" starting at 16384 & is 49131 Bytes. Only 40325 Bytes, starting at 24700 are required. On side two of the tape is a program which can only be loaded after side one is completed. S.J.Nutting has included a routine to load the side 2 on its own (very interesting effects...try it and see). Transfer side 1 as follows;

1. Type in then RUN this program:

```
10 CLEAR 65535: FOR J=65100 TO 65135: READ A: POKE J,A: NEXT J
20 DATA 243,221,33,0,37,17,235,191,55,62,255,205,86,5,48,241,
    33,1,227,17,1,254,1,133,157,237,184,33,123,96,34,178,92,205,
    183,17
```

```
30 RANDOMIZE USR 65100
```

2. Play in the MAX tape (side 1) AFTER the second Header. When all Loaded, Spectrum will Clear, but the Data includes a new Clear which ensure program part required is safe above RAMTOP
3. Enter these POKES POKE 24700,0: POKE 30464,4: POKE 30465,6: POKE 30468,4: POKE 30469,6: POKE 30584,48
4. Save to Microdrive by SAVE*"m";1;"MAXc" CODE 24700,40325
5. Basic Loader for this is;


```
10 CLEAR 24669: LOAD*"m";1;"MAXc" CODE 24700: RUN USR 59392
```

Save it to Microdrive by SAVE*"m";1;"MAX" LINE 10

For side 2, enter then RUN this program, then play in all side 2

```
10 CLEAR 65535: FOR J=65140 TO 65158: READ A: POKE J,A: NEXT J
20 DATA 49,255,255,221,33,0,64,17,110,190,55,62,255,205,86,5,
    195,0,128
30 RANDOMIZE USR 65140
```

SPY HUNTER TRANSFER by JAMES ANDREW MASTERS of ALRESFORD,HERTS

This is a Speedlock program so a Back-Up copy must first be made using 007-DEPULSER.

1. Make a False Header for 49151 Bytes.
2. Wind Back-Up copy to be just after the Basic section, then enter CLEAR 24575: LOAD "" CODE 16384
3. Play in the False Header you made, then play in rest of the Back-Up tape.....ignore error message when it's Loaded in.
4. Enter the following (MUST use line numbers 1 & 2 as shown);

```
1 FOR J=65500 TO 65511: READ A: POKE J,A: NEXT J: RANDOMIZE
USR 65500
```

```
2 DATA 33,255,230,17,207,238,1,0,135,237,184,207
```

This is a "Mover" program to make room for the Microdrive. RUN it, and ignore error message.

5. Enter CLEAR 26575

6. Enter SAVE*"m";1;"SPYHUNTERc" CODE 26576,34560

7. Type in this Loader program;

```
10 LOAD*"m";1;"SPYHUNTERc" CODE 30000
```

```
20 FOR J=65000 TO 65013: READ A: POKE J,A: NEXT J: RANDOMIZE
USR 65000
```

```
30 DATA 33,48,117,17,0,91,1,0,135,237,176,195,0,192
```

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

STARSTRIKE II TRANSFER by S.J.NUTTING of HISTON,CAMBRIDGE

The method of transfer is rather complicated and I regret that a full explanation is far too long to print. Method follows;

1. Type in this program:

```
10 CLEAR 57999: FOR A=58000 TO 58013: READ N: POKE A,N: NEXT A
```

```
20 DATA 221,33,241,227,17,62,28,55,62,255,205,86,5,201
```

```
30 RANDOMIZE USR 58000
```

```
40 FOR A=65266 TO 65279: READ N: POKE A,N: NEXT A
```

```
50 DATA 221,33,0,64,17,62,28,55,62,255,205,86,5,243
```

```
60 FOR A=65472 TO 65505: READ N: POKE A,N: NEXT A
```

```
70 DATA 33,206,255,17,0,64,1,20,0,237,176,195,0,64,33,0,255,
17,220,255,1,64,159,237,184,33,155,96,34,178,92,195,183,17
```

```
80 FOR A=65266 TO 65279: READ N: POKE A,N: NEXT A
```

```
90 DATA 221,33,0,64,17,62,28,55,62,255,205,86,5,243
```

```
100 IF INKEY$<>"L" THEN GO TO 100
```

```
110 RANDOMIZE USR 65266
```

2. Wind STARSTRIKE II tape to start of the long Headerless part.

3. RUN program just entered, then play in long Headerless part.

4. When finished you should see no Loading lines on Border. Now rewind Starstike tape back to start of the Long Headerless part, press L then play in Headerless part again. When Loaded Spectrum will NEW.

5. Enter the following;

```
10 FOR A=65266 TO 65279: READ N: POKE A,N: NEXT A
```

```
20 DATA 49,191,95,33,156,96,17,192,95,1,64,159,237,176,
62,195,50,255,255,175,205,155,34,195,119,238
```

```
30 POKE 61267,62: POKE 61268,99: POKE 61269,0
```

6. RUN the above, then press NEW then Enter.

7. Save to Microdrive by; SAVE*"m";1;"STARc" CODE 24732,40795

8. Basic Loader for the game is as follows;

```
10 CLEAR 24731: LOAD*"m";1;"STARc" CODE 24732: RUN USR 65501
```

(Save to Microdrive by; SAVE*"m";1;"STAR" LINE 10)

AUTOFILE ROUTINE by FRANK NELDER of HAMBURG,W GERMANY

This program is a form of RUN routine and is easily transferred to another cartridge at the touch of a button. The program also allows machine-code to be Loaded into alternative address if so desired....(useful for Dissassembling, etc).

```
10 ERASE "m";1;"zzzz": ERASE "m";1;"run"
```

```
20 OPEN # 5;"m";1;"zzzz": CAT # 5;1: CLOSE # 5
```

```
30 OPEN # 6;"m";1;"zzzz"
```

```
40 LET C=0: LET W=C: DIM A$(26,10)
```

```
50 FOR J=1 TO 26: INPUT # 6;X$: LET C=C+1: IF C=1 THEN LET Y$=X$
```

```
60 IF X$="run" " OR C<3 THEN NEXT J
```

```
70 IF X$="" THEN INPUT # 6;Z$: CLOSE # 6: SAVE*"M";1;"RUN"
```

```
LINE 200: GO TO 200
```

```
--> 80 IF X$(1)="*" THEN GO TO 100 <---see note about this
```

```
90 LET W=W+1: LET A$(W)=X$
```

```

100 NEXT J
200 CLS: BORDER 1: PAPER 1: INK 7: CLS
210 PRINT PAPER 2;"      CARTRIDGE: ";Y$;"      "
220 PRINT TAB 8;"(still ";Z$;"K free)"
230 LET C=0: PRINT: PRINT
240 LET C=C+1: IF A$(C, TO 2)="      " THEN GO TO 300
250 PRINT PAPER 2; BRIGHT 1; CHR$(64+C); PAPER 1; BRIGHT 0;"      ";
                                A$(C),: GO TO 240
300 PRINT # 0; PAPER 6; INK 0; "      Press Letter to LOAD
      (Press 0 to Generate new List) "
310 PAUSE 0: LET W$=INKEY$: IF W$="0" THEN GO TO 400
320 IF CODE W$<97 OR CODE W$>95+C THEN GO TO 310
330 POKE 23658,0: INPUT "Is it CODE program (y or n)";C$: IF C$=
      "n" THEN LOAD*"M";1;A$(CODE W$-96): STOP
340 INPUT "Start Addr (0=Normal Addr)";A: IF A=0 THEN LOAD*"m";
      1;A$(CODE W$-96)CODE: STOP
350 LOAD*"M";1;A$(CODE W$-96)CODE A: STOP
400 CLS # : PRINT PAPER 2; INK 7; FLASH 1; AT 9,0; "SAVEing the
      new program(s)....please wait": GO TO 10

```

Save to Microdrive by; SAVE*"m";1;"run"
 Line 80 is optional. If you include it, and prefix all machine-code parts of programs with an *, then listing on screen will be only the main title part of your games/programs. (Useful if you Load most programs by their main title). To use initially, Load this "run" program, put a cartridge with programs on into drive 1, press RUN and Enter. First time it is used it will be slow as it ensures zzzz & run not already on the cartridge. To transfer to another cartridge at a later date, Load this RUN from any of your cartridges, put cartridge it is to be transferred to into drive 1, and press 0 key to generate a new List for this cart.

BACK-UP COPIER TECHNIQUES

In the early days before Headerless-Files, Header-Readers could be used to get the start & length of machine code and the programs temporarily moved higher in memory to Save them. Early Headerless Files had been basically normal programs without the Header. This causes no problem now as a "False" header can be made, but such things hadn't been thought about then. The Spectrum reads the first Byte off a tape and if it is a zero it know it is a Header being Loaded in. If a 255, then it knows it is the main Block of the program (which is normally after the Header). Early Headerless Files had this 255 code at the start. Later it was discovered that a Headerless File could be Saved with ANY code value between 0-255. A copier has to be capable of accepting ANY code type and making the Copy have the same Code. This can be done by entering the ROM Load routine at 1369.

NEMONIC	HEX	COMMENTS
LD IX,26000	DD219065	This fairly simple method will accept
LD DE,65535	11FFFF	ANY Code type. The Code loads into addr
SCF	37	26000 and program Bytes go into 26001
EX AF,AF'	08	onwards. Message such as PLAY YOUR TAPE
CALL 1369	CD5905	and PRESS C TO COPY could be done by
LD (25998),IX	DD228865	Basic. Note that final IX value is put
RET	C9	into a safe location for later use.
SAVE ROUTINE SHOWN BELOW THE LINE.		
LD HL,(25998)	2A8865	Set HL=Last IX value.
LD BC,26002	019265	Set BC=IX value (+2 to ensure correct)
XOR	AF	XOR used to ensure carry not set.
SBC HL,BC	ED42	HL=True Length of program Loaded in.
PUSH HL	E5	Transfer HL to DE by PUSHing onto Stack,
POP DE	D1	then POPing off Stack into DE register.
LD IX,26001	DD219165	to start of the Bytes of the program.


```
LD A,(26000) 3A9065 Set A register to Code type Loaded in.
CALL 1218 CDC204 CALL the ROM Save routine.
RET C9 Return.
```

Whilst this program will copy any normal speed program, provided it is not too long, it only copies a part at a time. Multisection copiers are obviously more complicated as copier has to keep a record of all the various parts read in. (Sorry, far too long to include a method in MDX). Full 48K sections would give a problem as it leaves no room for the copier. Method used is to make the copier all machine code and put it at top of the memory. Example 48K copier routine could be as follows;

```
LD IX,65535 DD21FFFF This "pre-Loader" part reads only 1 Byte
LD DE,1 110100 which would be the Code type, and stores
SCF 37 it in 65535 for later used. Rewind Tape
EX AF,AF' 08 to start of the block, then all Block is
CALL 1369 CD5905 played in.
```

```
DI F3 Disables Interrupts as overwriting Vars.
LD IX,16224 DD21603F area. Note that we are actually reading
LD DE,49152 1100C0 into the ROM area....doesn't matter as a
SCF 37 full 48K starts with a screen$ and only
EX AF,AF 08 means a few extra dots and lines will be
CALL 1369 CD5905 on the initial screen when copy Loaded.
PUSH IX DDE5 Last IX value stored on "stack"
DI F3 Ensure Interrupts still disabled.
```

```
LD A,254 3EFE This is standard method to make program
IN A,(191) DBBF cycle until Enter key is pressed.
RRA 1F
JR NC -7 30F9
```

```
POP HL E1 Get last IX from off the stack into HL
LD BC,16226 01623F BC=Start IX+2 (to ensure correct length)
XOR AF Ensure carry not set.
SBC HL,BC ED42 HL=Length (could be less than 48K
PUSH HL E5 Transfer length to DE register by using
POP DE D1 stack
LD IX,16225 DD21613F Set IX to start of the Data Loaded in.
LD A,(65535) 3AFFFF Code type to A register.
CALL 1218 CDC204 Call ROM Save routine.
JP 0 C30000 Crash-out.
```

This copier could be used to copy ANY normal speed program part at a time. Copier would have to be loaded into high memory after doing a CLEAR 65476; the copier going into 65477 onwards. Allow 100 Bytes for Stack, therefore LOWEST Rom start address if full 48K to be Loaded is 16224. Program loads over the screen & screen would look a mess. When all loaded, press Enter key to Save out. Spectrum will Reset when Save done. Could be made check length, and restart if program not too long. Another method is to make it a "double-Load" type with first Load being used to just calculate the length and therefore start address in the ROM area required. Copier would then be able to copy upto the maximum possible of 65279 Bytes. (007-SPY was the first copier capable of copying such a length). Fast programs & "pulsating" types meant copiers again had to be updated.

Anyone with a 16K to 48K upgrade kit? Contact Frank Grimwood at 25 Castle Road, Rayleigh, Essex, SS6 7QD.

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