

SPECIAL OFFER
Spectravideo 318
for £99.95
see page 88

OCTOBER 1984 85p

WHICH MICRO & SOFTWARE REVIEW

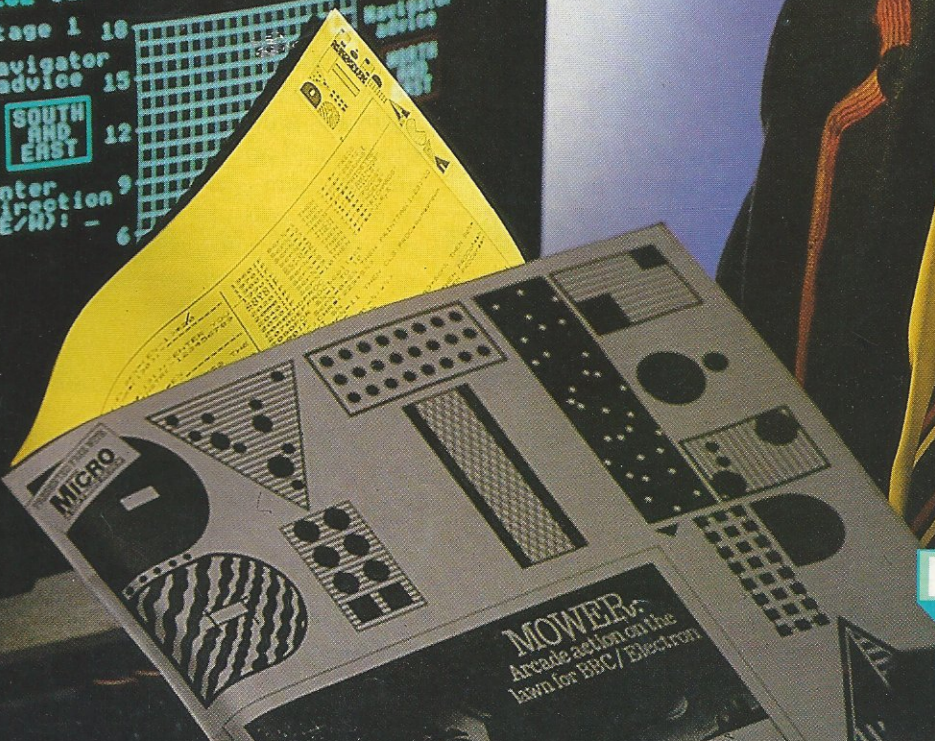
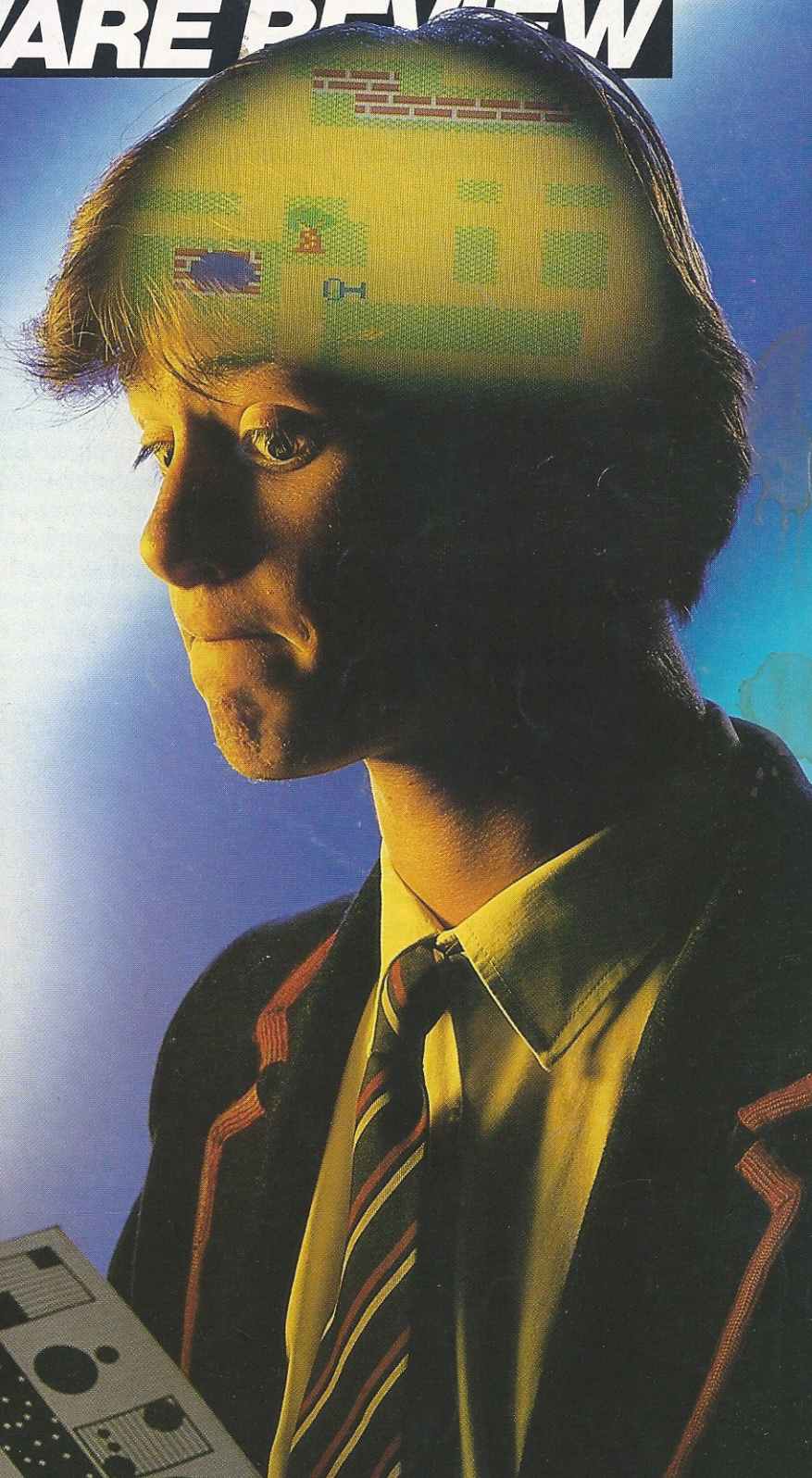
AN EMAP PUBLICATION U.S.A. \$2.95 Germany DM5.85

Educational
**SOFTWARE
SPECTACULAR**

- ALSO -

Free Inside
'BYTE PACK'

A 16-page pull-out of programs for
the Spectrum, Commodore, Dragon, Amstrad,
BBC, Memotech, Texas, QL and Electron.



Which Micro and Acorn Computers'
DESIGN-A-PROGRAM
COMPETITION
Over £1000 of prizes


```

840 RETURN
899 REM Dump to screen subroutine
900 AD=VAL ("£"+A$):EA=VAL ("£"+E$)
920 IF AD>EA THEN 1060
930 H$=HEX$(AD):L=LEN(H$)
940 PRINT " ";RIGHT$("0000"+RIGHT$(H$,L-1),4);"
945 J=EA-AD+1:IF J>8 THEN J=8
950 FOR I=1 TO J
960 CN=PEEK(AD):H$=HEX$(CN):L=LEN(H$)
970 PRINT " ";RIGHT$("00"+RIGHT$(H$,L-1),2);
980 AD=AD+1
990 NEXT I
1000 PRINT
1010 K$=KEY$:IF K$=CHR$(24) THEN 1060
1030 IF K$=" " THEN GET K$:IF K$=CHR$(24) THEN 10
60
1050 GOTO 920
1060 RETURN
1099 REM Printer machine code listing routine
1100 AD=VAL ("£"+A$):EA=VAL ("£"+E$)
1120 IF AD>EA THEN 1260
1130 H$=HEX$(AD):L=LEN(H$)
1140 LPRINT RIGHT$("0000"+RIGHT$(H$,L-1),4);" ";
1145 J=EA-AD+1:IF J>8 THEN J=8
1150 FOR I=1 TO J
1160 CN=PEEK(AD):H$=HEX$(CN)
1165 L=LEN(H$)
1170 LPRINT " ";RIGHT$("00"+RIGHT$(H$,L-1),2);
1180 AD=AD+1
1190 NEXT I
1200 LPRINT
1210 K$=KEY$:IF K$=CHR$(24) THEN 1260
1230 IF K$=" " THEN GET K$:IF K$=CHR$(24) THEN 12
60
1250 GOTO 1120
1260 RETURN
1500 END

```

Add Commands — Memotech

Most Memotech fans will be aware that the machine comes with a free text adventure euphemistically called 'The Operator's Manual'. For those that have managed to decipher the various sections of the manual, it should be apparent that whilst exceptional in certain areas (Assembler, Monitor, Logo graphics, Sprites and Noddy) the MTX series falls short in some others, after all 24K ROM can only hold so much.

Over the next few issues BYTE PACK will feature a set of short machine code utilities, from MTX expert Eric Roy of Kilmarnock, that will patch up certain discrepancies.

Before moving onto the utilities themselves, readers should note that as code cannot be assembled at a given address, at the end of each listing you will be required to use the PANEL's M(ove) command. This relocates your code in the last two pages of the BASIC program area (#BE00 to #CE00).

Listing 1 — 'OLD' enables the user to restore a BASIC program

that has been NEW'd or lost after the two reset keys have been pressed. The program is quite simple containing two sets of routines. The first set (SAVEB, SAVESV) copies the first 8 bytes of a BASIC program along with the system variables connected with BASIC, Noddy, and arrays to the safe locations #BE88 to #BE9C. The second set (RESTB, RESTSV) restores these bytes to their original location. To test the program you should enter LET O=USR(48640) to call the first two routines and then reset or NEW. Finally, to restore the program call the second set by entering LET O=USR(48707).

Before testing, however, code should be save and then moved. Owners of 512 and 500 models should move their code from #4007 ending #408C and from #8007 ending #808C respectively to #BE00.

A final note for MTX500 users, change the LD HL, (#4000), LD HL, (#4002) etc in the SAVEB and RESTB parts of the listing to LD HL, (#8000), LD HL, (#8002) etc.

Listing 2 — 'MERGE' allows you to append a subroutine (SUN) onto the body of a program (MAIN). Whilst this method is

somewhat circuitous, the process is invaluable for those who like to build up a library of generalised routines and so avoid duplicating their efforts on some later game or application.

Having typed in the program and with 'OLD' resident in memory, load the subroutine to be appended, enter CLEAR as a direct command and then enter LET O=USR(48800). The last instruction will call the routine SUB which uses 'OLD' to copy across the system variables and then moves a copy of your subroutine up in memory starting at #C400. This area is, in fact, reserved for the storage of variables, but as it is unlikely that any will be created whilst merging it can be safely overwritten. In any case, to be on the safe side #C400 instead of #C000 is used, which still allows programs of over 10K to be merged.

Now load the 'MAIN' program. Enter LET=USR(48820) to call the three routines MAIN, NONOD and MERGE. The first checks to see whether there are any Noddy pages to be merged and if so relocates them at the end of 'SUB'. The second moves both 'SUB' and any Noddy pages down to the end of 'MAIN'. The third adds the system values stored by OLD to those set when loading 'MAIN' to produce the new variables for the combined programs.

Again, before testing, ensure that the utility is saved and that it is moved from #4007, #4083 (MTX512) or #8007, #8083 (MTX500) to #BEA0.

```

5 REM **** BYTE PACK ****
6 REM **** OCT 1984 ****
100 CODE
803B SAVEB: LD HL, (£4000) ; SAVEB saves the
first 8 bytes of basic.
803E LD (£BE88),HL ; Store for 8 byte
s £BE88 to £BE8F.
8041 LD HL, (£4002)
8044 LD (£BE8A),HL
8047 LD HL, (£4004)
804A LD (£BE8C),HL
804D LD HL, (£4006)
8050 LD (£BE8E),HL
8053 SAVESV: LD HL, (£FAA4) ; Save Top of Nodd
y.
8056 LD (£BE90),HL ; Store for system
variables £BE90 to £BE9D.
8059 LD HL, (£FAA7) ; Save Top of Curr
ent Basic Page.
805C LD (£BE92),HL
805F LD HL, (£FAAA) ; Save Bottom of B
asic.
8062 LD (£BE94),HL
8065 LD HL, (£FAAC) ; Save Top of Each
Basic Page.
8068 LD (£BE96),HL
806B LD HL, (£FACC) ; Save Top of Arra
ys.
806E LD (£BE98),HL
8071 LD HL, (£FACF) ; Save Baselin.
8074 LD (£BE9A),HL
8077 LD HL, (£FAD6) ; Save Pqtop.
807A LD (£BE9C),HL ; All system varia

```

Listing 3 — 'INTERUPTS' (his spelling, not ours) not only makes 'OLD' available at the touch of a key but also corrects the manual (which fails to point out that bit 7 along with either 4,5 or 6 in the INTFFF variable must also be set before interrupts are vectored through the USRINT location).

The routine causes the computer to interrupt what it's doing 64 times every second to check to see whether a key has been pressed. This is achieved by taking the value of the last-key-pressed system variable (#FD7C) and comparing it with the key code of the keys to be tested. In this case the computer checks to see whether the Space Bar has been pressed and if so performs SAVEB and SAVESV routines in 'OLD'. Next it checks F1 and restores any NEW'd program. Unfortunately, F1 will not work on a reset as this turns the interrupts off (default).

Key F2 will also switch the interrupts off and must be used prior to loading a program as the USRINT location is also saved and may not contain the address of the KEYS interrupt routine.

Keys F3 and F4 will be covered in the next issue which will include routines to renumber program lines and to save and load code directly as opposed to via a BASIC program. In the meantime #404B to #4051 can be omitted if desired.

One last point: before testing, the code should be moved from #4007 ending #4053 (#8007 ending #8053) to #BF60.

bles now saved.

```

807D RET ; End of save routine.
807E RESTB: LD HL, (EBE88) ; Restore 8 basic
bytes.
8081 LD (E4000),HL
8084 LD HL, (EBE8A)
8087 LD (E4002),HL
808A LD HL, (EBE8C)
808D LD (E4004),HL
8090 LD HL, (EBE8E)
8093 LD (E4006),HL
8096 RESTSV: LD HL, (EBE90) ; Restore system v
ariables saved.
8099 LD (EFAA4),HL
809C LD HL, (EBE92)
809F LD (EFAA7),HL
80A2 LD HL, (EBE94)
80A5 LD (EFAAA),HL
80A8 LD HL, (EBE96)
80AB LD (EFAAC),HL
80AE LD HL, (EBE98)
80B1 LD (EFACC),HL
80B4 LD HL, (EBE9A)
80B7 LD (EFACF),HL
80BA LD HL, (EBE9C)
80BD LD (EFAD6),HL ; All system varia
bles restored.
80C0 RET

```

Symbols:

```

SAVEB 803B SAVESV 8053
RESTB 807E RESTSV 8096

```

```

5 REM **** BYTE PACK ****
6 REM **** OCT 1984 ****
100 CODE

```

```

803B INTON: LD A, E03 ; Code for JP (Jump).
803D LD (EFA98),A ; USERINT locatio
n.
8040 LD HL, EBF93 ; Interrupts vecto
red to this address = KEYS.
8043 LD (EFA99),HL ; USERINT +1,2 =
Interrupt vector address.
8046 LD A, (EFD5E) ; INTFFF system v
ariable.
8049 OR E9F ; Set bits 4 & 7.
804B LD (EFD5E),A ; INTERRUPTS ON.
804E RET
804F INTOFF: LD A, (EFD5E) ; INTFFF.
8052 AND E0F ; Bits 4 & 7 reset.
8054 LD (EFD5E),A ; INTERRUPTS OFF.
8057 RET
8058 REN1: LD HL, E64 ; First line number=100.
805B LD (EBF1E),HL ; Line number loc
ation.
805E JR STEP
8060 REN9: LD HL, E2328 ; First line numb
er=9000.
8063 LD (EBF1E),HL
8066 STEP: LD A, E0A ; Step between lines=10.
8068 LD (EBF1D),A ; Step location.
806B JP EBF20 ; Jump to renumber routin
e.
806E KEYS: LD A, (EFD7C) ; Last key presse
d system variable.
8071 CP E49 ; Is it SPACE BAR.
8073 JP Z, EBE00 ; Yes save system variabl
es in OLD.
8076 CP E48 ; Is it key 'F1'.
8078 JP Z, EBE43 ; Yes restore NEW'ed prog
ram system variables.
807B CP E46 ; Is it key 'F2'.
807D JR Z, INTOFF ; Yes switch inte
rupts off.
807F CP E43 ; Is it key 'F3'.
8081 JR Z, REN1 ; Yes renumber program fr
om 100.
8083 CP E41 ; Is it key 'F4'.
8085 JR Z, REN9 ; Yes renumber program fr
om 9000.
8087 RET

```

Symbols:

```

INTON 803B INTOFF 804F
REN1 8058 STEP 8066
REN9 8060 KEYS 806E

```

```

5 REM **** BYTE PACK ****
6 REM **** OCT 1984 ****
100 CODE

```

```

803B SUB: CALL EBE00 ; Store system variables
of 'SUB'.
803E LD HL, (EFAAA) ; HL=Start address
s of 'SUB'.
8041 LD DE, E0400 ; DE=Destination
address of 'SUB'.
8044 LD BC, (EFACC) ; BC=Length of 'S
UB'.
8048 LDIR ; Move 'SUB' from HL to D
E.
804A LD (EBE86),DE ; Store end address
of 'SUB' after move.
804E RET ; NOW LOAD 'MAIN' PROGRAM
.
804F MAIN: LD HL, (EFAA4)
8052 LD DE, (EFAA7)
8056 AND A
8057 SBC HL, DE ; HL=Length of any Noddy
pages in 'MAIN'.
8059 JR Z, NONOD ; If no Noddy pages then
jump, else
805B PUSH HL
805C POP BC ; BC=Length of 'MAIN' Nod
dy.
805D LD HL, (EFAA7) ; HL=Start address
s of 'MAIN' Noddy.
8060 LD DE, (EBE86) ; DE=End of 'SUB'
program.
8064 LDIR ; Move 'MAIN' Noddy to en
d of 'SUB'.
8066 LD (EBE86),DE ; Store new end a
dress of 'SUB'.
806A NONOD: LD HL, (EBE86) ; HL=End address
of 'SUB'.
806D LD BC, E0400 ; BC=Start address
s of 'SUB'.
8070 PUSH BC ; Save start address.
8071 AND A
8072 SBC HL, BC ; HL=Length of 'SUB'.
8074 PUSH HL
8075 POP BC ; BC=Length of 'SUB'.
8076 POP HL ; HL=Start address of 'SU
B'.
8077 LD DE, (EFAA7) ; DE=End of 'MAIN'
basic.
807B LDIR ; Move 'SUB' to end of 'M
AIN'.
807D MERGE: LD HL, (EBE90) ; Calculate 'NEW'
system variables.
8080 LD DE, (EFACC)
8084 ADD HL, DE
8085 PUSH HL
8086 LD (EFAAC),HL ; 'NEW' Top of ea
ch basic page.
8089 LD HL, (EFAA7)
808C LD DE, (EFAAA)
8090 AND A
8091 SBC HL, DE
8093 PUSH HL
8094 LD BC, (EBE92)
8098 ADD HL, BC
8099 LD (EFAA7),HL ; 'NEW' Top of cu
rrent basic page.
809C POP HL
809D LD BC, (EBE9C)
80A1 ADD HL, BC
80A2 LD (EFAD6),HL ; 'NEW' PGTOP.
80A5 LD HL, (EFAA4)
80A8 AND A
80A9 SBC HL, DE
80AB LD BC, (EBE98)
80AF ADD HL, BC
80B0 LD (EFACC),HL ; 'NEW' Top of ar
rays.
80B3 POP HL
80B4 LD (EFAA4),HL ; 'NEW' Top of No
ddy.
80B7 RET

```

Symbols:

```

NONOD 806A MERGE 807D
SUB 803B MAIN 804F

```

EOT