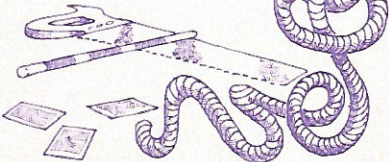


WHO HOLDS THE
THIRD KEY?
JUST ASK
THE DISPLAY
(WITH A C)

S
H
A
R
L

S
O
F
T



SHARPSOFT USER NOTES

Issue No. 12

C O N T E N T S

	<u>Page</u>
Editorial	2
Beginners Tutorial Guide to PASCAL	
Part 3 - Statements (cont'd from Part 2) and Simple input/output	3
Sharp MZ-80B	
Using a Modem with the MZ-80B - Chris Friedlander	8
Beethoven's Pour Elise - Ralph Birnbaum	13
Hardware Modification	
MZ-80K Gets Shugart Drives PART 3 - P. Sydenham	14
Member's Letters and Listings	
MZ-80A Listings	
Binary equivalents	25
Anagram program	26
Trail Layer a game for 2 Players	28
MZ-80K Listings	
Two-Stage Decay Chain Program	31
Holiday Route Planner	35
Golf Match Program	38

A vote of thanks to John Trippick
for the return of his marvellous artwork

© SHARPSOFT LTD. 1984

SHARPSOFT USER NOTES

Issue No. 12

Looking through the program listings presented in this Issue I noted that BASIC is the language which readers appear to do the majority of programming in. PASCAL and FORTH programmers please note we would value some contributions from you for future publication.

The theme this Issue is again applications and games. MZ-80B users will find the article on Using a Modem by Chris Friedlander both interesting and informative. The tiny music program from Ralph Birnbaum in Belgium is just a sample for future Issues.

A number of readers have expressed a wish that we include more facts, articles and programs for the MZ-80A, so we have an even split this Issue of A and K listings. The program "Trail Layer" by Mr. Simon Jones was written for the MZ-80A although Simon has also found that it will work correctly on the new MZ-700. If you are an MZ-80A user then please send us your programs and programming tips for publication.

The summer holiday season will soon be here again, along with holiday traffic jams, so Mr. Bent's Holiday Route Planner may prove more useful than you think. So too may his Golf Match Arranger which doubles as an everymans "Compound Gear Train Alternatives" program!! Many thanks Mr. Bent.

This Issue's FORTH contributions are included in our regular letters section.

Once again we feature our "Beginners Tutorial Guide to PASCAL" and Part 3 of Mr. Peter Sydenham's article on the MZ-80K disk drive interface.

The regular letters and listings sections of the user notes have been amalgamated at the back of this Issue. Our thanks to all the readers who have contributed to this Issue. Any prospective contributors, please send us a really good copy or tape of anything for publication, as poor copies cannot always be used for printing.

Issue 13 of the User Notes will be published towards the end of July 1984.

MIKE BRINSON

EDITOR

BEGINNERS TUTORIAL GUIDE TO PASCAL

PART 3

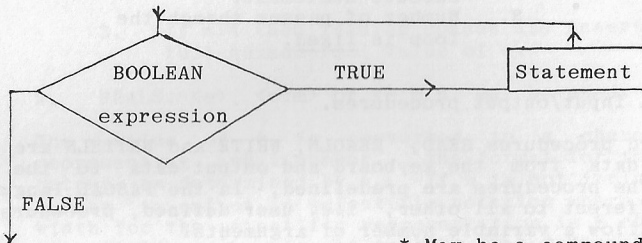
Statements (Cont'd)

and

Simple input/output

27. WHILE Statement.

WHILE <Boolean expression> DO Statement*;



* May be a compound Statement.

28. FOR Statement.

28.1 Increasing steps [increasing in unity steps]

FOR <control variable>:=expression 1 TO expression 2
DO Statement*;

28.2 Decreasing steps [decreasing in unity steps]

FOR <control variable>:=expression 1 DOWNTO expression 2
DO Statement*;

* May be a compound statement

NOTE the <control variable> must be a non-real scalar variable.

FOR statement examples.

1. MONTH:=6; DAYSINMONTH:=30
FOR DATE:=1TO DAYSINMONTH DO

WRITELN ('Date',MONTH:2, ' ',DATE:2);
2. TRAIN:=0;
FOR YEAR:=1983 DOWNTO 1900 DO
BEGIN
 READ(RAIN);
 TRAIN:=TRAIN+RAIN
END;

29. Summary of Looping Statements Properties.

REPEAT/UNTIL	<ol style="list-style-type: none"> 1. Multiple statements in loop. 2. Execute statements <u>then</u> Text exit condition. 3. Number of passes through the loop is variable.
WHILE...DO	<ol style="list-style-type: none"> 1. One statement in loop. 2. Check exit condition then execute statement. 3. Number of passes through the loop is variable.
FOR...TO/DOWNTO	<ol style="list-style-type: none"> 1. One statement in loop. 2. Check exit condition then execute statement. 3. Number of passes through the loop is <u>fixed</u>.

30. PASCAL input/output procedures.

The standard procedures READ, READLN, WRITE and WRITELN are used to input data from the keyboard and output data to the VDU screen. The procedures are predefined, in the PASCAL language, and are different to all other, i.e. user defined, procedures in that they allow a variable number of arguments.

30.1 WRITE

```
WRITE (P1, P2, P3....Pn); is the same as
BEGIN
  WRITE(P1); WRITE(P2); WRITE(P3);
  .....WRITE(Pn)
END;
```

The parameters (Pn) may be one of the following forms:-

<e>, <e:m>, <e:m:n> or <e:m:H>

where e, m and n are expressions and H is a literal constant.

Cases.

1. Integers. <e> or <e:m> may be used.

The value of the expression e is converted to a character string with a trailing space appended to the end of the string. The length of the string may be increased (by padding with leading spaces) by specifying an integer value for m. The integer m specified the total number of characters to be output. If m is not sufficient for e to be output in full, or m is not present, then e is written out in full, with a trailing space, and m is ignored. Also note that, if m is set to be the length of e without the trailing space then no trailing space will be output.

Hisoft PASCAL also includes a second form of formatted output for integers - this form allows integers to be output in hexadecimal notation.

This type of output uses <e:m:H>

Cases.

1. m=1 or m=2.

Then (e MOD 16)^m is output in a width of exactly m characters.

2. m=3 or m=4.

Then the value of e is output in hexadecimal in a width of 4 characters.

3. If m>4 then leading spaces are inserted before the full hexadecimal value of e.

2. REALS. <e>, <e:m> or <e:m:n> may be used.

The value of e is converted to a character string representing areal number. the format for the real value output is determined by n. If n is not included then the number is output in scientific notation where the mininum width for the scientific representation is 8 characters. If the field width m is set to less than 8 then the full width of 12 characters is always output. An example will help demonstrate the format rules:-

```
WRITE (-1.23E10:m);
```

m	yields	
7		-1.23000E+10
8		-1.2E+10
9		-1.23E+10
10		-1.230E+10
11		-1.23000E+10
12		-1.23000E+10
13		-1.23000E+10

If the form <e:m:n> is used then a fixed-point representation of the number e is output with n specifying the number of decimal places to be output.

3. CHARACTERS and STRINGS

Either <e> or <e:m> may be used. In this case the character or string of characters will be output in a minimum field width of 1 (for chars) or the length of the string (for strings). Note leading spaces are inserted if m is sufficiently large.

4. BOOLEANS

Either <e> or <e:m> may be used. In this case "TRUE" or "FALSE" will be output depending on the value of e-- note a minimum value of m=4 (TRUE) or m=5 (FALSE) is assumed.

30.3 WRITELN

WRITELEN outputs a new line - this is equivalent to
 WRITE (CHR(13)). A linefeed character is also output.
 WRITELN (P1, P2, P3....Pn); is equivalent to
 BEGIN
 WRITE (P1, P2, P3...Pn);
 WRITELN
 END;

Note:- WRITE and WRITELN can only be used for non-structured types; for example

WRITE(RECORD1); is not allowed if Variable RECORD1 is a RECORD type. It is, of course possible to output individual components of a structured type - see later notes on structured types.

30.3 READ

The procedure READ is used to enter data frm the keyboard. READ does this via a buffer held in the Hisoft compiler run-time package.

The run-time buffer is initially empty (except for an end of line market). Any access to this buffer takes place through a text window. We can access one character at a time through the buffer window. If the text window is positioned over an end of line marker, before a read operation is terminated, a new line of text will be read into the buffer from the keyboard.

READ (V1, V2, V3...Vn); is equivalent to
 BEGIN
 READ(v1); READ(V2); READ(V3);
 READ(Vn)
 END;

where V1...Vn are variables of type character, string, integer or real.

Cases.

1. Variable V is type character.

In this case READ(V) reads a character from the input buffer and assigns it to V. See page 20 of the Hisoft PASCAL manual for details.

2. Variable V is a string type.

A string of characters may be read using READ. In this case a series of characters will be read until the number of characters defined by the string has been read. See page 21 of the Hisoft PASCAL manual for details.

3. Variable V is of type integer.

In this case a series of characters which represent an integer is read. The integer is assigned to variable V. Automatic conversion of the characters to internal integer format takes place. See page 21 of the Hisoft PASCAL manual for details.

4. Vis of tape real.

Here, a series of characters representing a real number is read. See page 21 of the Hisoft PASCAL manual for details.

30.4 READLN

READLN (V1, V2, V3...Vn); is equivalent to

```
BEGIN
    READ(V1, V2, V3...Vn);
    READLN
END;
```

READLN simply reads in a new buffer from the keyboard.

30.5 The Hisoft PASCAL input/output routines recognise the various keyboard and display control codes detailed in section 0.0 of the Hisoft PASCAL manual. These are different for each implementation of the compiler. Consult the PASCAL implementation note for the codes for your machine i.e. MZ-80K, A or B.

30.6 An example program showing the input/output procedures.

```
PROGRAM MINMAX;
VAR
    I1, I2, I3 : INTEGER;
    AVERAG      : REAL;
BEGIN
    WRITELN('BEGIN CALCULATION');
    WRITELN('ENTER THREE INTEGERS ON ONE LINE');
    WRITELN('Separated by at least one ace');
    WRITELN('IF THE FIRST INTEGER IS ZERO');
    WRITELN('THE PROGRAM TERMIANTES');
    WRITELN('ENTER FIRST THREE NUMBERS');
    READLN ('I1, I2,I3);
    WHILE I1<> 0 DO
        BEGIN
            AVERAGE:=(I1+I2+I3)/3;
            WRITELN('THE AVERAGE IS ', AVERAGE);
            WRITELN('ENTER NEXT THREE NUMBERS');
            READLN(I1, I2, I3)
        END.
```

MORE PASCAL NEXT ISSUE

USING A MODEM WITH THE SHARP MZ-80B

by Chris Friedlander

12/05/84

Some time ago, I purchased an RS232C interface card for my MZ-80B as a means of using the machine to communicate with an analogue to digital converter. Supplied with the card, was yet another version of Disk Basic (SB-6511) and a very detailed technical manual. My application did not require any real delving into the intricacies of programming the Z80 SIO chips and I was able to ignore most of the instruction manual, which was just as well, as I did, and still do, find much of it difficult to understand.

Since then, I have been using one of the up market Sharp MZ-3541 business computers. This machine has a software controlled 'Terminal' mode, intended to link up with other machines via an RS232C link or a modem. Being curious, I acquired a modem and quickly discovered the world of Bulletin Boards and dial up databases. My curiosity aroused, I was naturally eager to hook up my MZ-80B to the nearest telephone.

It has often been a gripe of mine that the Sharp manuals launch straight into the technical meat, without much explanatory text for those of us with a less than perfect programming knowledge. A quick perusal of the RS232C manual, showed that, to use the device properly, requires a knowledge of assembly language. Although there are commands available to control the interface via Basic, as far as I can tell one can do little more than set the communications mode and pass string data in and out of the machine. To date, I have not seen any published hints on how to use an MZ-80B as a dumb terminal. I am in the process of obtaining a comprehensive modem control program to run under CP/M, but in the meantime decided to have a go myself, using the Sharp basic interpreter. The enclosed listing does work but is far from ideal. I have decided to offer it as a means of getting started, in the hope that some of the more able users in the group may be able to improve on it.

General Modem notes:

A modem is simply a device for converting data into audio frequency tones, suitable for transmission over the telephone network. There are very rigid standards imposed by British Telecom on the use of modems. These standards define the frequencies and signal levels used, as well as electrical safety considerations. I do not propose to go into detail about the rather long winded technical standards, as they don't concern us here.

There are two main categories of modem, direct connect or acoustic coupler. The direct connect type, as its name suggests, is connected directly to the telephone line. With an acoustic coupler, the telephone handset is placed in the device and the signals received or transmitted by means of the earpiece and mouthpiece respectively. The direct connect type scores heavily over the acoustic coupler in terms of lack of interference and reliability. Acoustic couplers can be subject to feedback over the phone line when communicating with a similar type at the remote end.

Technically speaking, it is relatively easy to build a modem oneself. The problem comes when you try to use it! It is ILLEGAL to connect ANY form of apparatus directly to the telephone line, which is not approved by British Telecom. I have seen several build it yourself kits and articles. They usually say the kit uses a "B.T. approved" transformer. This is not sufficient. If you build a modem, you must submit it to B.T. for technical approval. There is a charge for this service (Which often takes months) that makes the cost of a commercially built item seem insignificant. The situation with acoustic couplers is not so bad. As no connection to the line is involved, the only requirement, as far as I know, is that the frequencies used must be within the specifications.

I would question seriously, the wisdom of home construction. It is possible to buy an approved modem for a quite reasonable sum. I use a DaCom direct connect type which retails for just over 70 pounds. Genuine B.T. modems can be bought on the surplus market for similar, or lower prices.

Having decided on the modem type, the next question is which standard to use. Most of the dial up databases and Bulletin Boards in the U.K. operate on the 300 baud V21 standard system. Other services such as Prestel use a 1200/75 baud standard. Unless you want to access Prestel for which special hardware is available, 300 baud is the standard to use. Different data standards exist with the various users. As well as the baud rate, there are variations in the number of data bits (5-8), parity bits (Odd, Even or None) and stop bits (1, 1.5 or 2).

The program:

The program presented here, supports full duplex operation at 300 baud. A machine code program is loaded, starting at address \$F000. This provides for setting the SIO chips on the RS232C interface card to the relevant data format. Single character input and output routines are provided for both channel A and B although only channel B is used in this application. The Basic program is thus able to send and receive characters via calls to specific machine code routines. Incidentally, this information is provided in the Sharp RS232C interface manual and is used here in a condensed form.

Most standard communications software assumes that you can respond to the normal ASCII character codes. The MZ-80B only follows this standard between decimal 32 and 126. This can cause problems when codes outside this range are received. ASCII code 127 (Delete) and 10 (Line feed) will print a bent arrow and light up your shift lock key!! The software must therefore filter out the codes which the MZ-80B treats differently and modify them accordingly. Unfortunately this takes time and using Basic, the program is only just able to keep up with incoming data. In normal ASCII the codes below decimal 32 are assigned as follows:-

00 = NUL	09 = HT	17 = DC1	25 = EM
01 = SOH	10 = LF	18 = DC2	26 = SUB
02 = STX	11 = VT	19 = DC3	27 = ESC
03 = ETX	12 = FF	20 = DC4	28 = FS
05 = ENQ	13 = CR	21 = NAK	29 = GS
06 = ACK	14 = SO	22 = SYN	30 = RS
07 = BEL	15 = SI	23 = ETB	31 = US
08 = BS	16 = DLE	24 = CAN	

In addition, decimal 127 = delete and all codes above 127 are ignored. All other codes are as in the Sharp manuals. Once characters are received, they are poked into a large memory buffer for later recall. When they are stored and decoded they are displayed on the screen. The keyboard is then scanned for a character. If one is present, it is recoded if necessary and sent out to the interface card. Line feeds are added to any carriage returns sent. The control port is scanned at various times to see if characters are available for reception. The whole process then loops back until interrupted by the break key. A point to note: The buffer is a little under 24K in size. When full, it will return to the beginning and over-write the existing data, without giving any warning, so keep your sessions short if you wish to retrieve the data.

When the session is finished, the memory image can be recalled to the screen or sent to the printer. It is important to realise that this program is a very simple one. In particular, there is no communications protocol used so it is possible to receive garbled data at times. Also as presented, it is configured for 300 baud, 8 bit word, no parity and 1 stop bit. In order to change these parameters, it is necessary to alter some of the code in the first line of data statements. The details are set out in the interface manual. Because the program emulates a dumb terminal, it is not possible to transmit files or download software (Unless in standard ASCII for manual transcription).

I have not included a disk file option as this requires a form of protocol to halt input data while dumping the buffer to disk. I felt that as all hadn't been plain sailing so far, now was the time to offer this very basic introduction in the hope of generating enough interest. I would be interested to know if any other readers have come up with a better way of doing things.

Hardware notes:

To use this program you'll need, apart from the MZ-80B, an MZ-8BIO3 dual RS232C interface card, a modem and a minimum of one disk drive. With software modifications, it should be possible to work using tape basic, but I haven't tried it. Perhaps it is fair at this point to mention that, even at SharpSoft prices, the interface card isn't exactly cheap, partly due to its versatility.

On the MZ-8BIO3 interface card, each RS232C channel is terminated with a 9 pin D type connector. Different types of modem will have various types of connector.

The only signal lines that are used by this program are Transmit Data, Receive Data and Carrier Detect. These signals are on pins 2,3 and 7 when the card jumpers are set for Terminal Mode. If your modem has a complete set of data lines, by all means use them. If not, be sure to strap the RTS & CTS pins together (Nos. 4 & 5) and leave the DTR line (Pin 6) floating. Make sure that the baud rate switches are set for 300 (Sw 4 = ON) and the port selector switches set for ports B0 to B3 hex (Sw 1,2 and 5 = ON). I made up my own cable and am therefore not sure if the one available from Sharp is suitable or not.

The 25 pin standard RS232C D type connector:

This is the industry standard used on most computers and peripheral equipment, which has provision for control signals relating to a main and secondary channel. Most applications only use the 9 pins needed for the main channel, hence the use by Sharp of a 9 pin connector. As an aid to connecting equipment that uses a 25 pin D connector, here is a list of pin numbers relevant to the signal lines supported by the Sharp RS232C card.

Pin 1 = Protective Ground.	Pin 6 = Data Set Ready.
Pin 2 = Transmitted Data.	Pin 7 = Signal Ground.
Pin 3 = Received Data.	Pin 8 = Rec. Line Sig. Detect.
Pin 4 = Request to Send.	Pin 20 = Data Terminal Ready.
Pin 5 = Clear to Send.	All other pins are unused.

Please note that the meanings of some of these signal lines are reversed, depending whether the peripheral device is used for input or output. This explains the need for cross wired leads in some cases, although on the MZ-80B this can be avoided by changing the jumper positions on the interface card, as described in the manual.

Telephone numbers:

There are several private as well as commercially run Bulletin Boards in various parts of the country. In addition it is possible to use dial up ordering services available from some retail outlets. The list below is only intended as a start, if you contact one of these there will usually be information available about most of the other sources. It is as well to find out the transmission standard of unknown services beforehand, in order to configure your system. Also, watch your phone bill if contacting someone in a remote area. I have deliberately left out several numbers of overseas Bulletin Boards as well as those normally restricted to commercial users. All information is subject to change.

TBBS London.....	01 348 9400	Rewtel.....	0277 232628
Blandford Board.	0258 54494	Forum 80 London...	01 902 2546
Maptel.....	0702 552941	Mailbox Liverpool.	051 428 8924
CABB.....	01 631 3076	CBBS London.....	01 399 2136
Distel.....	01 679 1888	Forum 80 Hull.....	0482 859169

Program listing:

```

100 REM MODEM CONTROL PROGRAM FOR SHARP MZ-80B, USING AN MZ-8BIO3
    RS232C CARD
110 REM WRITTEN IN SHARP BASIC SB6510 OR 9B6511
120 REM C. FRIEDLANDER MAY 1984, BASED ON INFORMATION IN SHARP
    SERIAL
130 REM INTERFACE MANUAL
140 DATA 18,10,10,04,44,05,EA,03,E0,0E,B1,06,09,21,00,F0,ED,B3,0E,B3
150 DATA 06,09,21,00,F0,ED,B3,3E,03,D3,B1,3A,08,F0,F6,01,D3,B1,3E,03
160 DATA D3,B3,3A,08,F0,F6,01,D3,B3,C9,00,00,DB,B1,0F,30,FB,3E,01,D3
170 DATA B1,DB,B1,E6,70,32,32,F0,DB,B0,32,33,F0,C9,DB,B3,0F,30,FB,3E
180 DATA 01,D3,B3,DB,B3,E6,70,32,32,F0,DB,B2,32,33,F0,C9,DB,B1,CB,57
190 DATA 28,FA,3A,33,F0,D3,B0,C9,DB,B3,CB,57,28,FA,3A,33,F0,D3,B2,C9
200 DATA END
210 CONSOLE C80:PRINT TAB(20);"WAIT - Loading interface control
    program":PRINT:ON ERROR GOTO 6000
220 DIM X(30):LIMIT $F000:P=15*4096
230 FOR J=0 TO 9:X(J)=J:NEXT:FOR J=0 TO 5:X(17+J)=J+10:NEXT

```

```

240 READ X$:IF X$="END" THEN 270
250 J=16*X(ASC(MID$(X$,1,1))-48)+X(ASC(MID$(X$,2,1))-48)
260 POKE P,J:P=P+1:GOTO 240
270 POKE 37000,255
280 USR($F009)
290 PRINT CHR$(6):PRINT TAB(28);"TERMINAL EMULATION PROGRAM"
300 PRINT:PRINT"Select one of the options below :-"
310 PRINT:PRINT"1) Communicate via Modem."
320 PRINT:PRINT"2) Read memory based Terminal Log."
330 PRINT:PRINT"3) Print out memory based Terminal Log."
340 PRINT:PRINT"4) Delete memory based log."
380 PRINT:PRINT"5) Help."
390 PRINT:PRINT"6) Terminate session."
400 PRINT:PRINT"Enter choice (1 to 6):-"
410 GET ANS:IF ANS=0 THEN 410
420 IF (ANS<1)+(ANS>6) THEN 290
430 IF ANS=6 THEN PRINT CHR$(6):END
440 ON ANS GOSUB 1000,2000,3000,4000,5000
450 GOTO 290
460 RETURN
1000 PRINT CHR$(6):PRINT TAB(3);"Terminal Mode (300 Baud, 8 Bits,
    1 Stop, No Parity) - Ready to receive data":PRINT:PRINT:X=37000
1010 INP@ 179,S
1020 IF (S=100)+(S=108) THEN 1100
1030 USR($F04A):I=PEEK($F033)
1040 IF I<128 THEN POKE X,I:X=X+1
1050 IF X=61001 THEN X=37000
1060 IF I=127 THEN PRINT CHR$(7);
1070 IF I=10 THEN PRINT CHR$(13)+CHR$(1)
1080 IF (I>126)+(I<32) THEN 1100
1090 PRINT CHR$(I);
1100 GET KB$:IF KB$="" THEN 1010
1110 O=ASC(KB$)
1120 IF O=11 THEN 1160
1130 IF O=7 THEN O=127
1140 IF O=13 THEN POKE $F033,13:USR($F06C):POKE $F033,10:USR($F06C)
    :GOTO 1010
1150 POKE $F033,O:USR($F06C):GOTO 1010
1160 POKE X+1,255:RETURN
2000 PRINT CHR$(6):R=37000
2010 E=PEEK(R):IF R>61000 THEN 2100
2020 IF E=255 THEN 2100
2030 GET KB$:IF KB$=CHR$(11) THEN GOSUB 2080
2040 IF E=127 THEN PRINT CHR$(7);
2050 IF E=10 THEN PRINT CHR$(13)+CHR$(1)
2060 IF (E<127)*(E>31) THEN PRINT CHR$(E);
2070 R=R+1:GOTO 2010
2080 GET KB$:IF KB$="" THEN 2080
2090 RETURN
2100 PRINT:PRINT:PRINT"END OF STORED TEXT [ Bytes used =" ;R-37000;
    " Bytes free =" ;61000-R; "]"
2110 PRINT:PRINT"Press any key to return to main menu : "
2120 GET KB$:IF KB$="" THEN 2120
2130 RETURN
3000 PRINT CHR$(6):PRINT"MAKE SURE PRINTER IS READY - Press P to
    print, any other to quit"
3010 GET KB$:IF KB$="" THEN 3010
3020 IF (KB$="p")+(KB$="P") THEN PRINT:PRINT"Ready to print":GOTO 3040
3030 RETURN
3040 FOR X=37000 TO 61000
3050 L=PEEK(X)

```

```

3060 IF L=255 THEN PRINT/P:PRINT/P"* END OF FILE *":RETURN
3070 IF L>126 THEN 3110
3080 IF L=13 THEN PRINT/P CHR$(10)
3090 IF L<9 THEN 3110
3100 PRINT/P CHR$(L);
3110 GET KB$:IF KB$=CHR$(11)THEN PRINT/P:PRINT/P"* PRINT STOPPED *"
:RETURN
3120 NEXT X
3130 PRINT/P:PRINT/P"* FILE FULL UP - PRINT ABANDONED *":RETURN
4000 PRINT CHR$(6)
4010 PRINT"Are you sure ? (Y or N)"
4020 GET K$:IF K$="" THEN 4020
4030 IF (K$="y")+(K$="Y") THEN 4050
4040 RETURN
4050 PRINT"CLEARING MEMORY":FOR T=1 TO 100:NEXT T
4060 POKE 37000,255:RETURN
5000 PRINT CHR$(6)
5010 PRINT"This program is designed to allow the MZ-80B to emulate a
dumb terminal."
5020 PRINT:PRINT"The RS232C interface is set for 300 Baud, 8 Bit
word, No parity, 1 Stop bit."
5030 PRINT:PRINT"In the terminal mode, use the BREAK key to return
to the main menu when you"
5040 PRINT:PRINT"have finished transmitting and receiving."
5050 PRINT:PRINT"The break key can also be used to stop and start
reading or listing the log."
5060 PRINT:PRINT:PRINT"During the session, all data in both
directions is stored in a buffer, in"
5070 PRINT:PRINT"memory. This buffer is about 24K in size and should
be adequate for most"
5080 PRINT:PRINT"sessions. If the buffer fills, no warning is given
and subsequent data is"
5090 PRINT:PRINT"entered at the start of the buffer, overwriting the
existing text."
5100 PRINT:PRINT:PRINT"Press any key to return to the main menu : "
5110 GET K$:IF K$="" THEN 5110
5120 RETURN
6000 PRINT CHR$(6):PRINT "***** ERROR - RETURNING YOU TO MAIN MENU
*****":USR($OEBE)
6010 FOR T=1 TO 2000:NEXT T:GOTO 290

```

Beethoven's Pour Elise for the MZ-80B

from

Mr. R. Birnbaum in Belgium

```

100 REM
200 REM
300 REM
400 REM
46000 TEMPO6:MUSIC"+E1+#D+E+#D+EB+D+CA3R1"
46001 MUSIC"C1EAB3R1E#GB+C3R1E+E+#D+E+#D+EB+D+CA3R1CEAB3R1E+CBA3"
46002 MUSIC"R1B1+C+D+E4G1+F+E+D4F1+E+D+C4E1+D+CB3R1"
46003 MUSIC"E1+EE+E", "+E3+#D1+E+#D+E+#D+E+#D+E+#D+E+#D+EB+D+CA3R1"
46004 MUSIC"C1EAB3R1E#GB+C3R1E+E+#D+E+#D+EB+D+CA3R1CEAB3R1E+CBA3R1"

```

HARDWARE MODIFICATION

PAGE 1

```

1      ;SHARP DISC INITIALIZE CODE
2      ;PREPARED USING DISASSEMBLER BY
3      ;R TANSWELL AND 'ZEN' ASSEMBLER
4      ;17 10 83 P SYDENHAM
5
6
7 3000 310030  COLD:  LD  SF,COLD           ;STACK STARTS HERE
8 3003 CD0900  CALL  CRLF
9 3006 119D30  LD   DE,TITLE       ;* SLAVE-DISK ETC.
10 3009 CD1500 CALL  MESSAGE
11 300C CDCB34  CLD1: CALL STPMOT
12 300F CD0900 CALL  CRLF
13 3012 11C230 LD   DE,DRVPRMT     ;DRIVE NO. ( ETC.
14 3015 CD1500 CALL  MESSAGE
15 3018 11CB33 LD   DE,BUFF2
16 301B CD0300 CALL  USER          ;GET ANSWERE TO PROMPT
17 301E 1A     LD   A,(DE)
18 301F FE1B   CP   1BH           ;BREAK KEY ?
19 3021 2BE9   JR   Z,CLD1        ;Z=YES GO BACK TRY AGAIN
20 3023 211B00 LD   HL,001BH
21 3026 19     ADD  HL,DE           ;SKIP OVER PROMPT
22 3027 7E     LD   A,(HL)         ;GET ANSWERE IF A CHAR.
23 3028 FE21   CP   21H           ;WAS IT !
24 302A CA0000 JP   Z,MONITOR      ;YES
25 302D FE0D   CP   0DH           ;CR ?
26 302F 2002   JR   NZ,CLD2        ;CR EQUIVALENT TO 1
27 3031 3E31   LD   A,31H         ;SO LOAD ACC. ASCII 1
28 3033 47     LD   B,A           ;KEEP ASCII
29 3034 E6F0   AND  0F0H          ;MASK
30 3036 FE30   CP   30H           ;CHECK IT WAS BETWEEN
31 3038 20D2   JR   NZ,CLD1        ;1 AND 4 INC.
32 303A 78     LD   A,B           ;AND FORM NUMERIC VALUE
33 303B E60F   AND  0FH
34 303D 3D     DEC  A             ;0 FOR DRIVE 1
35 303E FE04   CP   04H          ;1 FOR DRIVE 2
36 3040 30CA   JR   NC,CLD1       ;ETC.
37 3042 32C232 LD   (DVBUF),A     ;KEEP SELECTED DRIVE
38 3045 CD0900  CLD3: CALL  CRLF
39 3048 11DE30 LD   DE,VOLPRMT    ;VOLUME NO. ETC.
40 304B CD1500 CALL  MESSAGE
41 304E 11CB33 LD   DE,BUFF2      ;GET VOLUME NO.
42 3051 CD0300 CALL  USER         ;VIA USER ROUTINE
43 3054 1A     LD   A,(DE)
44 3055 FE1B   CP   1BH           ;BREAK ?
45 3057 CA0C30 JP   Z,CLD1
46 305A 210B00 LD   HL,000BH
47 305D 19     ADD  HL,DE
48 305E 110000 LD   DE,0000
49 3061 7E     LD   A,(HL)         ;GET ANSWERE
50 3062 FE0D   CP   0DH
51 3064 2B16   JR   Z,CLD6
52 3066 CDA032 CALL  KPCHK        ;CHECK FOR VALID CHAR.
53 3069 38DA   JR   C,CLD3       ;C=NOT VALID
54 306B 07     RLCA
55 306C 07     RLCA
56 306D 07     RLCA
57 306E 07     RLCA
58 306F 0604   LD   B,04H
59 3071 B7     CLD5: OR   A
60 3072 17     RLA

```

HARDWARE MODIFICATION

PAGE 2

```

61 3073 CB13          RL  E
62 3075 CB12          RL  D                ;ASSUME VOL NO. IS 123
63 3077 10F8         DJNZ CLD5           ;THEN
64 3079 23           INC  HL                ;DE ENDS UP HOLDING VOL.
65 307A 18E5         JR   CLD4                ;NO. IN PACKED BCD
66 307C 7A           LD  A,D                ;IE DE HOLDS 0123
67 307D FE02         CP   02H                ;CHECK VOL NO.<200
68 307F D24530       JF   NC,CLD3
69 3082 7B           LD  A,E
70 3083 CD7032       CALL CONVX          ;CONVERT BCD TO HEX
71 3086 14           INC  D                ;CHECK THAT VOL < 128 DEC
72 3087 15           DEC  D                ;IF NOT ABORT
73 3088 2804         JR   Z,CLD8
74 308A C664         ADD  A,64H
75 308C 18F9         JR   CLD7
76 308E FE80         CP   80H
77 3090 D24530       JF   NC,CLD3
78 3093 B7           OR   A
79 3094 CA4530       JF   Z,CLD3
80 3097 32CC32       LD  (K2CC),A        ;KEEP VOL NO. IN HEX
81 309A C3EE30       JF   START          ;NOW START PROG. PROPER
82 309D 2A20534C     TITLE: DB  "* SLAVE-DISK"
83 30A1 4156452D
83 30A5 4449534B
83 30A9 20494E49     DB  " INITIALIZE "
83 30AD 5449414C
83 30B1 495A4520
84 30B5 2837392F     DB  "(79/12/15) *"
84 30B9 31322F31
84 30BD 3529202A
85 30C1 0D           DB  0DH
86 30C2 44524956     DRVPRMT: DB  "DRIVE NO."
86 30C6 45204E4F
86 30CA 2E
87 30CB 28312D34     DB  "(1-4)[!-MONITOR]"
87 30CF 295B212D
87 30D3 4D4F4E49
87 30D7 544F52
88 30DA 5D203F       DB  "J ?"
89 30DD 0D           DB  0DH
90 30DE 564F4C55     VOLPRMT: DB  "VOLUME NO.?"
90 30E2 4D45204E
90 30E6 4F2E3F
91 30E9 0D           DB  0DH
92 30EA 454E44       ENDMESS: DB  "END"
93 30ED 0D           DB  0DH
94 30EE 3EC3         START:  LD  A,0C3H        ;JP CODE
95 30F0 320B10       LD  (ERRESC),A     ;SET UP ERROR ESCAPE
96 30F3 21ED31       LD  HL,ERRTRF     ;ADDRESS ERROR TRAP
97 30F6 220C10       LD  (ERRESC+1),HL
98 30F9 CDCB34       CALL STPMOT        ;STOP MOTOR
99 30FC 3E00         LD  A,00H          ;SET UP DATA THAT WILL
100 30FE 32C332      LD  (TKBUF),A     ;BE ACCESSED VIA
101 3101 3E01         LD  A,01H          ;(IX+D) FOR DISC READ
102 3103 32C432      LD  (SCTBF),A     ; TRACK 00
103 3106 216400      LD  HL,0100        ;SECT 01
104 3109 22C532      LD  (LNGBF),HL    ;256 BYTES
105 310C 21CB33      LD  HL,BUFF2      ;DESTINATION FOR READ
106 310F 22C732      LD  (LOCBF),HL
107 3112 DD21C232    LD  IX,DVBUF       ;SET UP IX REG.

```

HARDWARE MODIFICATION

PAGE 3

```

108 3116 CDD134      CALL DSCREAD          ;READ DISC INTO 33CB
109 3119 3ACB33      LD A,(BUFF2)         ;GET 1ST BYTE OF CODE
110 311C FEC3        CP OC3H              ;READ IN IS IT C3
111 311E 200F        JR NZ,MAIN          ;NZ=NOT MASTER DISC
112 3120 CDCB34      CALL STPMOT          ;STOP MOTOR
113 3123 CD0900      CALL CRLF
114 3126 111F32      LD DE,INHIBIT        ;INHIBIT INIT. ETC.
115 3129 CD1500      CALL MESSAGE
116 312C C30030      JP COLD              ;GO BACK TO SQUARE 1
117 312F 211632      LD HL,ETS           ;SET NEW ERROR
118 3132 220C10      LD (ERRESC+1),HL     ;ESCAPE ADDR.
119 3135 0680        LD B,80H             ;NOW CLEAR 33CB-344B
120 3137 21CB33      LD HL,BUFF2
121 313A 22C732      LD (LOCBF),HL
122 313D AF          XOR A
123 313E 77          LD (HL),A
124 313F 23          INC HL
125 3140 10FC        DJNZ MAIN2
126 3142 DD21C232    LD IX,DVBUF         ;RESET IX REG.
127 3146 CDD734      CALL DISCINIT
128 3149 3E00        LD A,00H            ;RESET BUFFERS ACCESSED
129 314B 32C332      LD (TKBUF),A       ;BY (IX+D)
130 314E 3E01        LD A,01H            ;READY FOR CHECK READ
131 3150 32C432      LD (SCTBF),A       ;OF 1ST 4 TRACKS
132 3153 210020      LD HL,2000H        ;READ 2000H BYTES
133 3156 22C532      LD (LNGBF),HL
134 3159 DD21C232    LD IX,DVBUF         ;RESET IX REG.
135 315D CDCE34      CALL DVERIFY        ;DUMMY READ FOR VERIFY
136 3160 300F        JR NC,MAIN3         ;NC=DISC OK
137 3162 CDCB34      CALL STPMOT
138 3165 CD0900      CALL CRLF
139 3168 113932      LD DE,NOUSDSC      ;NO USABLE DISC ETC.
140 316B CD1500      CALL MESSAGE        ;BUM DISC
141 316E C30030      JP COLD             ;BACK TO SQUARE 1
142
143 ;NO FAULTS ALLOWED IN READING
144 ;ANY OF SECTORS IN FIRST 4 TRACKS
145 ;THE REST OF THE DISC IS NOW
146 ;READ AND IF A FAULTY SECTOR IS
147 ;FOUND THE BIT MAP FOR THE
148 ;WHOLE TRACK IS LOADED TO
149 ;GIVE THE IMPRESSION THAT THE
150 ;TRACK IS FULL OF DATA
151 ;THIS THEN PREVENTS ANY DATA
152 ;FROM BEING WRITTEN TO THIS TRACK
153 ;IN FUTURE
154
155 ;AT THE VERY END THE BIT MAP IS
156 ;WRITTEN TO THE DISC
157 ;NOTE SECTOR COUNT IS ALSO
158 ;CORRECTED FOR ANY TRACKS THAT
159 ;HAVE BEEN MADE NON-AVAILABLE
160 3171 AF          MAIN3: XOR A
161 3172 32CB32      LD (K2CB),A
162 3175 06FE        LD B,0FEH
163 3177 21CD32      LD HL,K2CD
164 317A 77          MAIN4: LD (HL),A
165 317B 23          INC HL
166 317C 10FC        DJNZ MAIN4
167 317E 3E04        LD A,04H            ;START CHECK AT TRACK 4

```

HARDWARE MODIFICATION

PAGE 4

```

168 3180 32C332      LD   (TKBUF),A
169 3183 3E01        LD   A,01H          ;SECTOR 1
170 3185 32C432      LD   (SCTBF),A
171 3188 212003      LD   HL,0800        ;READ 1 TRACK A GO
172 318B 22C532      LD   (LNGBF),HL    ;IE 8*256 BYTES
173 318E DD21C232    LD   IX,DVBUF       ;RESET IX REG.
174 3192 CDCE34      MAIN5: CALL DVERIFY        ;DUMMY READ
175 3195 3837        JR   C,MAIN7        ;CARRY=FAULT
176 3197 3AC332      MAIN6: LD   A,(TKBUF)   ;TRACK
177 319A 3C          INC  A
178 319B 32C332      LD   (TKBUF),A
179 319E FE46        CP   46H            ;END OF DISC ?*****
180 31A0 38F0        JR   C,MAIN5        ;CARRY=NOT END
181 31A2 3E00        LD   A,00H
182 31A4 32C332      LD   (TKBUF),A     ;TRACK 0
183 31A7 3E0F        LD   A,0FH         ;SECTOR OF IE BIT MAP
184 31A9 32C432      LD   (SCTBF),A
185 31AC 216400      LD   HL,0100       ;256 BYTES TO WRITE
186 31AF 22C532      LD   (LNGBF),HL
187 31B2 21CB32      LD   HL,K2CB
188 31B5 22C732      LD   (LOCBF),HL
189 31B8 DD21C232    LD   IX,DVBUF       ;RESET IX REG.
190 31BC CDD434      CALL DSCWRITE      ;DISC WRITE
191 31BF CDCB34      CALL STPMOT
192 31C2 CD0900      CALL CRLF
193 31C5 11EA30      LD   DE,ENDMESS    ;END
194 31C8 CD1500      CALL MESSAGE
195 31CB C30C30      JP   CLD1          ;BACK TO START ALL DONE
196 31CE 2ACD32      MAIN7: LD   HL,(K2CD) ;
197 31D1 110A00      LD   DE,0010
198 31D4 19          ADD  HL,DE
199 31D5 22CD32      LD   (K2CD),HL
200 31D8 3AC332      LD   A,(TKBUF)     ;NO. OF FAULTY TRACK
201 31DB D604        SUB  04H            ;ALLOW FOR 1ST MAPPED TRA
202 31DD 07          RLCA               ;BEING TRACK 4
203 31DE 5F          LD   E,A           ;COMPUTE PLACE IN BIT MAP
204 31DF 1600        LD   D,00H
205 31E1 21CF32      LD   HL,K2CF
206 31E4 19          ADD  HL,DE
207 31E5 36FF        LD   (HL),OFFH     ;FILL TRACK AREA OF
208 31E7 23          INC  HL            ;BIT MAP
209 31E8 36FF        LD   (HL),OFFH
210 31EA C39731      JP   MAIN6         ;CONTINUE
211 31ED 310030      ERRTRP: LD   SP,COLD    ;NEW STACK
212 31F0 3A0810      LD   A,(ERRCOD)
213 31F3 FE36        CP   36H
214 31F5 CA2F31      JP   Z,MAIN
215 31F8 FE32      ET1: CP   32H
216 31FA 2005      JR   NZ,ET2
217 31FC 114832      LD   DE,NORDY      ;NO READY MESSAGE
218 31FF 180C      JR   ET4
219 3201 FE36      ET2: CP   36H
220 3203 2005      JR   NZ,ET3
221 3205 115132      LD   DE,UNFORM     ;UNFORMAT MESSAGE
222 3208 1803      JR   ET4
223 320A 116032      ET3: LD   DE,DSCDAT ;DISK DATA ERR
224 320D CD0900      ET4: CALL CRLF
225 3210 CD1500      CALL MESSAGE
226 3213 C30030      JP   COLD
227 3216 310030      ET5: LD   SP,COLD    ;NEW STACK

```

PAGE 5

```

228 3219 3A0810          LD  A,(ERRCOD)
229 321C C3F831          JF  ET1
230 321F 20494E48 INHIBIT: DE  " INHIBIT"
230 3223 49424954
231 3227 20494E49          DE  " INITIALIZE"
231 322B 5449414C
231 322F 495A45
232 3232 20455252          DE  " ERROR"
232 3236 4F52
233 3238 0D              DE  ODH
234 3239 4E4F2055 NOUSDSC: DE  "NO USABLE"
234 323D 5341424C
234 3241 45
235 3242 20444953          DE  " DISK"
235 3246 4B
236 3247 0D              DE  ODH
237 3248 4E4F2052 NORDY:  DE  "NO READY"
237 324C 45414459
238 3250 0D              DE  ODH
239 3251 554E464F UNFORM:  DE  "UNFORMAT"
239 3255 524D4154
240 3259 20455252          DE  " ERROR"
240 325D 4F52
241 325F 0D              DE  ODH
242 3260 4449534B DSCDAT:  DE  "DISK DATA"
242 3264 20444154
242 3268 41
243 3269 20455252          DE  " ERROR"
243 326D 4F52
244 326F 0D              DE  ODH
245
246
247
248
249           ; CONVERT BCD TO HEX
250
251 3270 C5              CONX:  PUSH BC
252 3271 4F              LD   B,A
253 3272 E60F           AND  OFH
254 3274 4F              LD   C,A
255 3275 78              LD   A,B
256 3276 E6F0           AND  OFOH
257 3278 07              RLCA
258 3279 07              RLCA
259 327A 07              RLCA
260 327B 07              RLCA
261 327C 47              LD   B,A
262 327D 79              LD   A,C
263 327E 05              CONX1: DEC  B
264 327F FAB732         JP   M,CONX2
265 3282 C60A           ADD  A,0AH
266 3284 C37E32         JP   CONX1
267 3287 C1              CONX2: POP  BC
268 3288 C9              RET
269
270
271
272 3289 C5              PUSH BC
273 328A 0600           LD   B,00H
274 328C D60A           K28C: SUB  0AH

```

PAGE 6

```

275 328E 3803      JR   C,K293
276 3290 04       INC  B
277 3291 18F9     JR   K28C
278 3293 C60A     K293:  ADD  A,0AH
279 3295 CB00     RLC  B
280 3297 CB00     RLC  B
281 3299 CB00     RLC  B
282 329B CB00     RLC  B
283 329D B0       OR   B
284 329E C1       POP  BC
285 329F C9       RET
286
287
288 32A0 C5       KPCHK:  PUSH  BC           ;CHECK FOR NUMERIC
289 32A1 E5       PUSH  HL           ;KEY PRESSED AND EXIT
290 32A2 01000A   LD   BC,0A00H     ;WITH ACC. HOLDING
291 32A5 21B832   LD   HL,NUMTAB    ;THE NUMBER
292 32A8 BE       KPCH1:  CP   (HL)         ;EXIT WITH CARRY
293 32A9 2004     JR   NZ,KPCH2     ;SET IF NON-NUMERIC
294 32AB 79       LD   A,C
295 32AC E1       POP  HL
296 32AD C1       POP  BC
297 32AE C9       RET
298
299
300 32AF 23       KPCH2:  INC  HL
301 32B0 0C       INC  C
302 32B1 05       DEC  B
303 32B2 20F4     JR   NZ,KPCH1
304 32B4 37       SCF
305 32B5 E1       POP  HL
306 32B6 C1       POP  BC
307 32B7 C9       RET
308
309
310
311
312               ;LIST OF VALID KEYS
313
314 32B8 30313233  NUMTAB:  DB   "0123456789"
315 32BC 34353637
316 32C0 3839
317
318               ;BUFFER USED FOR PARAMETERS REOD.
319               ;DURING DISC READ OR WRITE
320               ;ACCESSED VIA IX+---
321
322 32C2 00       DVBUF:  DB   00H
323 32C3 00       TKBUF:  DB   00H           ;TRACK BUFFER
324 32C4 00       SCTBF:  DB   00H
325 32C5 0000    LNGBF:  DB   00H,00H           ;LENGTH READ/WRITE
326 32C7 CB33    LOCBF:  DB   0CBH,33H
327 32C9 0000    DB   00H,00H
328 32CB 00       K2CB:  DB   00H
329 32CD 00       K2CC:  DB   00H
330 32CD 0000    K2CD:  DB   00H,00H
331 32CF 00000000 K2CF:  DB   00H,00H,00H,00H
332               DS   0FBH           ;ALL FULL OF ZEROS

```

HARDWARE MODIFICATION

PAGE 7

```

333 33CB 00      BUFF2:    DB    00
334              DS    OFFH
335
336 34CB C3F134  STPMOT:    JP    MOTOFF
337 34CE C37037  DVERIFY:   JP    DUMBREAD
338 34D1 C3F735  DSCREAD:   JP    READER
339 34D4 C3C236  DSCWRITE:  JP    WRITER          ;WRITES BIT MAP
340 34D7 C3F737  DISCINIT:  JP    FORMAT
341
342 34DA C5      STRTMOT:   PUSH  BC          ;START MOTOR
343 34DB 01F808  LD    BC,08F8H
344 34DE ED78    IN    A,(C)
345 34E0 010000  LD    BC,0000
346 34E3 0B      STMT1:    DEC    BC
347 34E4 00      NOP
348 34E5 00      NOP
349 34E6 78      LD    A,B
350 34E7 B1      OR    C
351 34E8 20F9    JR    NZ,STMT1
352 34EA 3E01    LD    A,01H
353 34EC 320210  LD    (MOTFLG),A
354 34EF C1      POP    BC
355 34F0 C9      RET
356
357 34F1 C5      MOTOFF:   PUSH  BC
358 34F2 CDB736  CALL  LNGDEL        ;CALL LONG DELAY
359 34F5 01F800  LD    BC,00F8H      ;TURN MOTOR OFF
360 34F8 ED78    IN    A,(C)
361 34FA AF      XOR    A            ;AND CLEAR DRIVE STORES
362 34FB 320210  LD    (MOTFLG),A    ;TO SHOW NONE SELECTED
363 34FE 320310  LD    (DSTR1),A
364 3501 320410  LD    (DSTR2),A
365 3504 320510  LD    (DSTR3),A
366 3507 320610  LD    (DSTR4),A
367 350A C1      POP    BC
368 350B C9      RET
369
370
371
372 350C CD1735  SKZERO:   CALL  SHTSTS
373 350F AF      XOR    A
374 3510 D3F9    OUT   (OF9H),A      ;CLEAR TRACK REG.
375 3512 320010  LD    (OPSTR),A
376 3515 D3FA    OUT   (OFAH),A      ;SEND SEEK ZERO CODE
377
378              ;SHORT STATUS CHECK
379
380 3517 C5      SHTSTS:   PUSH  BC          ;CHECK DRDY,CDRY,RQM
381 3518 010000  LD    BC,0000
382 351B DBF9    SHST1:   IN    A,(OF9H)
383 351D E603    AND   03H          ;MASK LEAVES CRDY,RQM
384 351F FE02    CP    02H          ;IS CRDY ONLY PRESENT
385 3521 2002    JR    NZ,SHST2     ;NZ=NO
386 3523 C1      POP    BC
387 3524 C9      RET
388
389 3525 0B      SHST2:   DEC    BC          ;KEEP TRYING
390 3526 78      LD    A,B
391 3527 B1      OR    C
392 3528 18F1    JR    SHST1

```

HARDWARE MODIFICATION

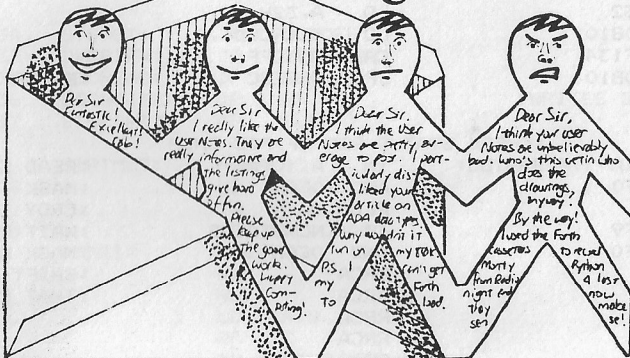
PAGE 8

```

393 352A C1          POP BC
394 352B 3E32       LD A,32H
395 352D 320810     LD (ERRCOD),A
396 3530 CDF134     CALL MOTOFF
397 3533 C30B10     JP ERRESC          :ERROR ESCAPE
398
399
400                  :CHECK STATUS REG OF FDC CHIP
401 3536 DBFA       STATUS: IN A,(OFAH)          :READ STATUS
402 3538 E6F0       AND OFOH          :MASK
403 353A 07         RLCA          :CRDY INTO CARRY FLAG
404 353B 30F9       JR NC,STATUS      :WAIT FOR CRDY
405 353D E6F0       AND OFOH          :MASK AGAIN
406 353F 0F         RRCA          :SHIFT STATUS BITS SUCH
407 3540 0F         RRCA          :THAT S3 BECOMES DBO
408 3541 0F         RRCA
409 3542 0F         RRCA
410 3543 E7         OR A
411 3544 C8         RET Z          :Z=ALL OK
412 3545 FE06       CP 06H          :ARE S1 & S2 ONLY SET
413 3547 2812       JR Z,ST3         :Z=YES
414 3549 FE0C       CP 0CH          :ARE S3 & S2 SET
415 354B 2004       JR NZ,ST1        :NZ=NOT S3 & S2
416 354D 3E32       LD A,32H          :ERR CODE
417 354F 180A       JR ST3
418 3551 FE04       ST1: CP 04H          :IS S1 ONLY SET
419 3553 2004       JR NZ,ST2
420 3555 3E36       LD A,36H          :ERR CODE
421 3557 1802       JR ST3
422 3559 3E29       ST2: LD A,29H          :ERR CODE
423 355B 320810     ST3: LD (ERRCOD),A
424 355E 37         SCF
425 355F C9         RET
426
427
428 3560 C5         PRMDRV: PUSH BC          :PRIMES DRIVE IN USE
429 3561 E5         PUSH HL
430 3562 3A0210     LD A,(MOTFLG)
431 3565 0F         RRCA          :IS MOTOR RUNNING
432 3566 D4DA34     CALL NC,STRTMOT
433 3569 DD7E00     LD A,(IX+00H)      :GET DRIVE NO.
434 356C E603       AND 03H          :MASK & FORM NEW CODE
435 356E F61C       OR 1CH          :SET TND,MOTOR,SELECT BIT
436 3570 320110     LD (DRVST),A      :KEEP DRIVE CODE
437 3573 E60F       AND OFH          :MASK OUT TND BIT
438 3575 47         LD B,A
439 3576 0EFB       LD C,OF8H
440 3578 ED60       IN H,(C)          :SEND DRIVE SELECT TO FDC
441 357A 3E32       LD A,32H
442 357C CDB736     PR1: CALL LNGDEL        :CALL LONG DELAY
443 357F 3D         DEC A          :TIME FOR MOTOR AND HEAD
444 3580 20FA       JR NZ,PR1
445 3582 010000     LD BC,0000
446 3585 DBF9       PR2: IN A,(OF9H)        :GET SHORT STATUS
447 3587 E607       AND 07H          :MASK UNUSED BITS
448 3589 FE06       CP 06H          :WAIT FOR CRDY & DRDY
449 358B 2017       JR NZ,PR5
450 358D DD4E00     LD C,(IX+00H)     :DRIVE NO.
451 3590 0600       LD B,00H          :CALC WHERE DRIVE
452 3592 210310     LD HL,DSTR1      :STORE IS

```

LETTERS page



Dear Sharpsoft,

I was most interested to read Mr. Rawson's hints on how to get the MZ-80A to accept a nul string. Here are some ideas for the same problem with the MZ-80K.

1. (Crude but effective).

Enter a double quote in response to the prompt for input: BASIC will accept this and return a nul string.

2. (More formal).

```
1000 POKE6350,0: Q$=CHR$(63)+SPC(1)+CHR$(34): POKE6350,34
1010 PRINTQ$: INPUT" ";A$
```

This displays the double quotes on the screen and leaves the cursor on it: pressing CR gives a nul string whilst anything keyed in overwrites the quotes.

3. (Best of all).

```
1000 POKE 7910,0: POKE 7922,0: POKE 7994,0: POKE 7995,0: POKE
7996,0
```

This allows BASIC to accept a nul string. to return to normal.

```
2000 POKE 7910,13: POKE 7922,13: POKE 7994,204: POKE 7995,154:
POKE 7996,31
```

I hope these will be of interest. Congratulations on S.U.N. I look forward to every issue!

N. ELLIS
WILTS

Dear Editor,

Sharpsoft Fig-FORTH on the MZ-80A

Having the revised tapes, that is the ones without the bug in the Line Editor on SCR # 12, I recently and somewhat belatedly started to learn FORTH. I bought "Starting FORTH" by Leo Brodie and used this together with a combination of back numbers of Sharpsoft User Notes. However I have come to a halt at page 66 of Brodie. This is because the Line Editor does not contain 'words' such as 'F' or any other string editing 'words'. However, using this Editor, I was able to enter the V2.0 editor from S.U.N. Issue 5, pages 4-6 (5.4-6) and the improved VLIST, ID., etc. from 7.44, but to my disappointment, in spite of the '-->' at the bottom of the VLIST program, it does not run onto the screens containing the editor V2.0. Nor does the V2.0 load properly. It always stops loading at the end of SCR # 10. If I then go on to load SCR # 11 it will not stop loading at the end of SCR # 12 in spite of the ';S'. If I then 'BREAK' and enter 'EDITOR', it still won't run properly. 'F' produces an error, so I am back to page 66.

Another criticism is that you never answered the letter on 5.41 about what to do if one got the endless 'PRINTER OFF' message produced by inadvertently touching INST while SHIFT is depressed. Also P. Bennett's new VLIST reverses the screen half way through (and if he had '2 SPACES' instead of 5 in the definition it would go on 2 screens.) The question of the dictionary words 2SWAP, 2OVER and 2DROP that are absent is mainly answered in the EDITOR V2.0 listing, but Brodie considered them important as they are only on page 52.

Are my main troubles due to trying to run the K version of the FORTH language tape on an A? Is it that my A has an 80 column conversion? Shall I wait until I can afford disk drives, and try again?

Meanwhile I very much enjoy the User Notes.

DR. C. MOON
SUSSEX

In the past I have used the MZ-80K FORTH on the MZ-80A without undue difficulty. I did however, notice that listing an empty screen caused display flicker. As far as I am aware V2.0 of the Screen editor works O.K. on the MZ-80K. I can only suggest that your 80 column modification is causing your problems. One point that should be noted is as follows - a FORTH screen is defined as 16 rows (numbered 0 to 15) of 64 characters per row. Hence if you type characters beyond column position 64 this will probably cause problems. Each screen consists of 16 x 64 = 1024 characters. However 16 x 80 = 1280 characters and characters beyond 1024 will probably be lost - hence a possible answer to why your "-->" is not found.

Perhaps other readers can suggest a solution to this problem?

EDITOR

Dear Sharpsoft,

MZ-80K + EPSON PRINTER MX-80F/T with GRAPHICS

Thank you for your letter I do hope you can help.

I am having difficulty printing 'Joined Graphics'. On the screen they are joined but when printed I cannot close the lines. I have tried several ways without success. I am trying to print a large letter 'A' full screen size.

Looking forward to your reply.

MRS. B. LAVENDER
MIDDX

The answer to your problem is to use the command Print/P "␣" to suppress the gap between lines, so that the graphics characters will match up. Obviously, the Print/P "␣" must be used before the graphics are printed. This particular solution only applies to your set-up, anyone else wishing to try this should refer to their printer manual for the appropriate codes needed.

SHARPSOFT

Dear Sharpsoft,

I should like to take this opportunity to say how much I appreciate the User Notes, which in their value for the Sharp computer far surpass any other publication I have seen.

I enclose two listings which you may find of interest for the Notes. One is a short program I have written to print out on my Sharp P5 printer a list of binary equivalents of numbers from 0 to 255 and from -1 to -127 and the other is a translation for the MZ-80A of an anagram program published in Electronics and Computing in April 1982 which uses as its base the Scrabble character set.

Having had something of a struggle to come to grips with FORTH, I have written out a child's guide to saving program screens:

SHARPSOFT FORTH
Routine for saving program screens

1. Load core FORTH tape.
2. Type in FORMAT definition and format a tape for the number of screens needed.
3. Load Editor tape.
4. Type EDITOR (CR).
5. Insert formatted tape.
6. Type 1 LIST (CR) and re-wind tape to starting position.
7. Type in screen data using appropriate Editor commands.
8. Type FLUSH. The screen response should be:
 UPDATING SCR # 1
 POSITION TAPE !
 TYPE CR WHEN READY
9. As the tape has already been positioned (step 6), press CR.
10. Operate the tape unit as directed by the screen reponses, as follows:

```

PLAY
PRESS STOP THEN CR
RECORD.PLAY
PRESS STOP THEN CR

```

11. The screen response should then be OK, indicating that the screen has been saved.
12. Return to step 6 for subsequent screens, using the appropriate screen numbers, and re-winding to end of previously saved screen.

BINARY NUMBERS

```

Sharp Basic SA-5510      1.1k
7 REM
8 REM---INTRODUCTION---
9 REM
10 PRINT "E";TAB(13);"*****BINARY NUMBERS":PRINTTAB(13);"*****"
20 PRINT "*****THIS PROGRAM PRINTS TABLES OF BINARY
30 PRINT "*****EQUIVALENTS OF DECIMAL NUMBERS
40 PRINT "*****PRESS A FOR THE TABLE FROM 0 TO 255
50 PRINT "*****PRESS B FOR THE TABLES FROM 0 TO 127 AND
55 PRINT "*****FROM -1 TO -127
60 GETP$:IFP$=""THEN60
70 IFP$="B"THEN200
77 REM
78 REM---LIST FROM 0 TO 255---
79 REM
80 CLR:M=0:N=204:A=1
90 FORK=MTONSTEP51:I=K:GOSUB300
100 DNAGOSUB400,410,420,430,440
110 A=A+1:NEXTK
120 M=M+1:N=N+1:IFM>50THEN140
130 A=1:GOTO90
140 K=255:GOSUB300:GOSUB440:GOTO10
197 REM
198 REM---LIST FROM 0 TO 127 AND FROM -1 TO -127---
199 REM
200 CLR:M=0:N=100:A=1
210 FORK=MTONSTEP25:I=K:GOSUB300
220 DNAGOSUB400,410,420,430,440
230 A=A+1:NEXTK
240 M=M+1:N=N+1:A=1:IFN<128THEN210
250 PRINT/P:PRINT/P:CLR:M=-1:N=-101:A=1
260 FORK=MTONSTEP-25:I=ABS(K)+128:GOSUB300
270 DNAGOSUB400,410,420,430,440
280 A=A+1:NEXTK
290 M=M-1:N=N-1:IFN<-127THEN10
295 A=1:GOTO260
297 REM
298 REM---S/R - CALCULATE BINARY NUMBERS---
299 REM
300 A$="":FORJ=1TO8
310 IFI/2<>INT(I/2) THENA$=A$+"1":GOTO330
320 A$=A$+"0"
330 I=INT(I/2):NEXTJ
340 B$="":FORJ=1TO8
350 B$=B$+RIGHT$(A$,1)
360 A$=LEFT$(A$,LEN(A$)-1)
370 NEXTJ:RETURN
397 REM

```

```

398 REM---S/R - PRINT TABLES---
399 REM
400 PRINT/PTAB(4);K;TAB(9);B#;:RETURN
410 PRINT/PTAB(19);K;TAB(24);B#;:RETURN
420 PRINT/PTAB(34);K;TAB(39);B#;:RETURN
430 PRINT/PTAB(49);K;TAB(54);B#;:RETURN
440 PRINT/PTAB(64);K;TAB(69);B#;:RETURN

```

ANAGRAMS

Sharp Basic SA-5510

2k

```

7 REM
8 REM INTRODUCTION
9 REM
10 PRINT "ANAGRAMS";TAB(16);"ANAGRAMS";TAB(56);"=====
20 PRINT "Make as many words as you can from seven
21 PRINTTAB(16);"letters
27 REM
28 REM SET UP ARRAYS AND VARIABLES
29 REM
30 DIMW$(98),L$(7)
35 A=18692:HS=0
40 DATA A,A,A,A,A,A,A,A,A,A,B,B,C
41 DATA C,D,D,D,D,D,E,E,E,E,E,E,E
42 DATA E,E,E,E,E,F,F,G,G,G,H,H
43 DATA I,I,I,I,I,I,I,I,J,K,L
44 DATA L,L,M,M,N,N,N,N,N,N,O
45 DATA O,O,O,O,O,P,P,Q,R,R
46 DATA R,R,R,R,S,S,S,S,T,T,T,T
47 DATAT,U,U,U,U,V,V,W,W,X,Y,Z
50 FORI=1TO98:READW$(I):NEXT
60 FORW=1TO1000:NEXT
100 TI$="000000":T=0:S=0:L$=""
107 REM
108 REM SELECT KEY LETTERS
109 REM
110 FORJ=1TO7
115 I=INT(RND(1)*98)+1:IFW$(I)="O"THEN115
120 L$(J)=W$(I):W$(I)="O":L$=L$+L$(J):NEXT
130 PRINT "E";TAB(16);L#;TAB(56);"-----
137 REM
138 REM PRINT OPTIONS, TIME AND SCORES
139 REM
140 S=0
200 PRINTTAB(5);"P"RESS 1 TO ACCEPT A WORD
210 PRINTTAB(5);"C"PRESS 2 TO CANCEL A WORD
220 PRINTTAB(5);"S"PRESS 3 TO STOP
230 PRINTTAB(5);"T"IME USED";TAB(16);T;" SECONDS
240 PRINTTAB(5);"S"CORE";S;TAB(20);"HIGHEST SCORE";HS
297 REM
298 REM PRINT LETTERS OR CONTROL NUMBERS
299 REM
300 X=4:Y=13
310 GETB$:IF(B$="")+ (PEEK(A)=1) THEN310
320 T=VAL(MID$(TI$,3,2))*60+VAL(RIGHT$(TI$,2))
330 CURSOR16,9:PRINTT;" SECONDS":IFT>300THEN400
340 NL=NL+1
350 IF(B$="1")*(NL<3) THEN500
360 IFB$="1" THEN600
370 IFB$="2" THEN700
380 IFB$="3" THEN800

```

```

390 CURSORX,Y:PRINTB#:X=X+1:GOTO310
397 REM
398 REM   END OF TIME
399 REM
400 FORJ=1TO3
410 CURSOR10,22:PRINTSPACE$(18):FORW=1TO50:NEXTW
420 CURSOR10,22:PRINT"END OF TIME":FORW=1TO50:NEXTW
430 NEXTJ:GOTO800
497 REM
498 REM   WORD TOO SHORT
499 REM
500 CURSOR4,22:PRINT"TOO SHORT - MINIMUM IS 3 LETTERS
510 FORW=1TO1000:NEXT
520 X=X-NL:CURSORX,Y:PRINTSPACE$(NL)
530 CURSOR4,22:PRINTSPACE$(32)
597 REM
598 REM   ACCEPT WORD AND CHECK FOR END OF GAME
599 REM
600 IF(X<11)*(Y<20)THENX=4:Y=Y+1:GOTO670
610 IF(X<11)*(Y=20)THENX=12:Y=13:GOTO670
620 IF(X<19)*(Y<20)THENX=12:Y=Y+1:GOTO670
630 IF(X<19)*(Y=20)THENX=20:Y=13:GOTO670
640 IF(X<27)*(Y<20)THENX=20:Y=Y+1:GOTO670
650 IF(X<27)*(Y=20)THENX=28:Y=13:GOTO670
660 IFX>27THENX=28:Y=Y+1
670 S=S+1:CURSOR10,11:PRINTS:IFS=30THEN690
680 NL=0:GOTO310
690 CURSOR5,22:PRINT"*** 30 WORDS - END OF GAME ***":GOTO800
697 REM
698 REM   DELETE WORD
699 REM
700 X=X-NL:CURSORX,Y:PRINTSPACE$(NL):X=X+1:GOTO310
797 REM
798 REM   END OF GAME - PRINT SCORE
799 REM
800 MUSIC"R5":PRINT"█";TAB(14);"█HIGHEST SCORE";TAB(20);S
810 PRINTTAB(10);"█HIGHEST SCORE IS";
820 IFS>STHEN840
830 HS=S
840 PRINTHS
897 REM
898 REM   PLAY AGAIN?
899 REM
900 PRINT"█DO YOU WANT ANOTHER GAME? (Y OR N)
910 Z$="":GETZ$:IF(Z$<>"N")*(Z$<>"Y")THEN910
920 IFZ$="Y"THEN100
930 PRINT"█:CURSOR10,10:PRINT"THANK YOU. GOOD-BYE.
940 PRINT"█":END

```

(Based on a game for the Video Genie by A A Hunting Horn and published in "Electronics and Computing, April 1982)

S.G.A. DOW
SURREY

MEMBERS' LETTERS & LISTINGS

```

650 PRINT"      HIT SPACE BAR TO CONTINUE
660 GET A$:IF A$<>" " THEN 660
670 PRINT"
680 PRINT"PLAYER 1 STARTS FROM THE TOP. Keys:-"
690 PRINT
700 PRINT"          8....Move up"
710 PRINT"          2....Move down"
720 PRINT"          4....Move left"
730 PRINT"          6....Move right"
740 PRINT"
750 PRINT"PLAYER 2 STARTS FROM THE BOTTOM. KEYS:-"
760 PRINT"
770 PRINT"          W....Move up"
780 PRINT"          X....Move down"
790 PRINT"          A....Move left"
800 PRINT"          D....Move right"
810 PRINT"
820 INPUT" Enter name of player one ";N1$:N1$=LEFT$(N1$,6)
830 INPUT" Enter name of player two ";N2$:N2$=LEFT$(N2$,6)
840 PRINT"
850 PRINT "          PRESS SPACE BAR TO START"
860 GET A$:IF A$<>" " THEN 860
870 GOTO 100
1000 PRINT"X=50
1010 CURSOR 0,23
1020 PRINT"
1030 PRINT"
1040 PRINT"
1050 PRINT"
1060 PRINT"
1070 PRINT"
1080 PRINT"
1090 PRINT"
1100 PRINT"
1110 PRINT"
1120 PRINT"
1130 PRINT"
1140 PRINT"
1150 PRINT"
1160 PRINT"
1170 PRINT"
1180 FOR P=1 TO 6
1190 PRINT:FDR J=1 TO 20:NEXT
1200 GOSUB 1230
1210 NEXT P :USR(71):GOTO 1240
1220 REM SUB
1230 X=X-1:POKE 4514,X:USR(68):RETURN
1240 PRINT"
1250 PRINT"          (C) Simon Jones 1984"
1260 PRINT
1270 PRINT"          Hit Space Bar When Ready"
1280 TI$="000000"
1290 IF TI$>"000040" THEN TI$="000000":RETURN
1300 GET A$:IF A$=" " THEN RETURN
1310 GOTO 1290

```

S. JONES
KENT

Dear Sir,

I enclose a program in BASIC for the MZ-80K for the Sharpsoft User Notes. It illustrates graphically the decay of a radioactive material and the build-up and decay of its radioactive 'daughter'. The only non-standard feature used in the program is the PRINT@ command. This may be replaced in standard SP-5025 by POKEing 4465 and 4466. Thus PRINT@ X,Y;"text" becomes POKE4465,X: POKE4466,Y: PRINT"text".

As usual I eagerly await the next issue of SUN. Your magazine is the best of its kind, its principal disadvantage being the long intervals between issues.

It would be interesting to know if the level of interest in the MZ-80K, A, B micros - as evidenced by the subscriptions and contributions to SUN - is on the increase or decrease. The impression obtained from the general monthly micro magazines is that it is on the decrease - this in spite of Practical Computing's recent introduction of a SHARP page (late last year PC rejected a program I had submitted on the grounds that it has been written for the MZ-80K! They asked for it to be rewritten for an IBM or a BBC micro!!!).

Keep the SUNs coming.

A. STEVENS
DERBY

At this time the level of interest in the MZ-80K, A and B personal micros is constant - the number of subscriptions to SUN have remained roughly the same for about 18 months. We anticipate this situation will not change for at least one further year. We will be continuing to support the MZ-80K, A and B micros by publishing these notes for at least one more year. We do already publish a separate set of User Notes for the Sharp MZ-700 and will in the future do our best to support any new machines developed by Sharp for the home user.

EDITOR

```

1 PRINT" "
2 PRINTTAB(9);"TWO-STAGE DECAY CHAIN"
3 PRINTTAB(9);"*****"
4 PRINT
5 REM   □ Alan Stevens
6 REM
7 REM   March 1984
8 REM   *****
9 REM
10 REM  A$ = picture of graphical axes
11 REM  LI$ = line for tabular output
12 REM
13 DL$=" "
14 A$=""
15 FOR Y=1 TO 19
16 A$=A$+CHR$(203)+DL$
17 NEXT
18 A$=A$+CHR$(205)
19 FOR X=1 TO 15
20 A$=A$+CHR$(209)
21 NEXT
22 LI$=CHR$(227)
23 FOR X=1 TO 37
24 LI$=LI$+CHR$(227)
25 NEXT
26 REM

```

MEMBERS' LETTERS & LISTINGS

```

27 REM R$(R),Y$(R) = labels
28 REM P(X),D(X) = data for tabular output
29 REM
30 DIM R$(1),Y$(1),P(30),D(30)
31 R$="Relative "
32 R$(0)="number of atoms.":R$(1)="activity. "
33 LB$="Half-lives"
34 Y$(0)="N"+DL$+"U"+DL$+"M"+DL$+"B"+DL$+"E"+DL$+"R"+DL$+" " +DL$+" "
35 Y$(1)="A"+DL$+"C"+DL$+"T"+DL$+"I"+DL$+"V"+DL$+"I"+DL$+"T"+DL$+"Y"
36 REM
37 REM F,Y1,Y2,Z# = constants
38 REM
39 F=38:Y1=4:Y2=42:Z#="0 "
40 GOSUB 150
41 REM
42 REM Obtain half-life data
43 REM
44 R=1:PRINT"@ "
45 PRINT:PRINT"NH = number of half-lives of parent"
46 PRINT:PRINT"radioactive material. Enter the number"
47 PRINT:PRINT"of these half-lives you want to consider"
48 INPUT"(where 0< NH <1500):";NH
49 IF (NH<=0)+(NH>1500) THEN 44
50 S$=STR$(NH)
51 REM
52 REM PX$ = X-axis label
53 REM
54 PX$=LEFT$(Z$,16-LEN(S$))+S$
55 REM
56 REM DT,TP,LT = constants needed for recursive calculations below
57 REM
58 DT=NH/30:TP=2^DT:LT=LN(TP)
59 REM
60 REM Obtain more half-life data
61 REM
62 PRINT:PRINT:PRINT"L = parent half-life/daughter half-life."
63 PRINT"Enter the size of this ratio you wish"
64 PRINT:INPUT"to consider (0< L <1500/NH):";L
65 IF (L<=0)+(L>1500/NH) THEN PRINT" ";SPC(39);"###":GOTO 64
66 S$=STR$(L*NH)
67 REM
68 REM DX$ = another X-axis label
69 REM SP = another constant for the recursion calculations
70 REM
71 DX$=LEFT$(Z$,16-LEN(S$))+S$
72 SP=TP^L
73 PRINT"@ "
74 REM
75 REM Set up initial conditions
76 REM
77 R=1-R
78 P=TP:S=SP:D=-LT*TP:Y1=Y1:Y2=Y2:A=R*L+1-R
79 REM
80 REM Draw and label axes of graphs
81 REM
82 PRINT"@";TAB(5);R$;R$(R)
83 PRINTTAB(9);"Parent";TAB(27);"Daughter";
84 PRINT@1,2;"1":PRINT@1,8;Y$(R):PRINT@1,21;"0":PRINT@3,2;A$
85 PRINT@21,2;"1":PRINT@21,8;Y$(R):PRINT@21,21;"0":PRINT@23,2;A$
86 PRINTTAB(3);PX$;TAB(23);DX$
87 PRINTTAB(6);LB$;TAB(26);LB$
88 REM

```

MEMBERS' LETTERS & LISTINGS

```

89 REM For each x-coordinate, calculate old and new y-coordinate
90 REM values, unplot old graphs (with RESET) and plot new graphs
91 REM (with SET).
92 REM
93 FOR X=7 TO 37
94 P=P/TP:S=S/SP
95 IF L=1 THEN D=D/TP+P*LT:GOTO 97
96 D=(P-S)/(L-1)
97 YP=Y2-INT(P*F+.5):YD=Y2-INT(L*D*F/A+.5)
98 RESETX,YP:RESETX+40,YD
99 YD=Y2-INT(A*D*F+.5)
100 P(X-7)=P:D(X-7)=A*D
101 SETX,YP:SETX+40,YD
102 NEXT X
103 REM
104 REM Display options available
105 REM
106 PRINT"A: ";LEFT$(R$(1-R),8);" E:end N:new data T:tabulate";
107 GET U$
108 IF U$="A" THEN PRINT"█":PRINTSPC(39):GOTO77
109 IF U$="E" THEN 146
110 IF U$="N" THEN 44
111 IF U$="T" THEN 116
112 GOTO 107
113 REM
114 REM Display data in tabular form
115 REM
116 PRINT"█":R$:R$(R)
117 PRINTLI$
118 PRINT"T/S";TAB(10);"PARENT";TAB(25);"DAUGHTER"
119 PRINTLI$
120 FOR X=0 TO 15
121 PRINTX;TAB(9);P(X);TAB(24);D(X)
122 NEXT
123 PRINTLI$
124 PRINT"T/S=timestep (timestep interval=NH/30)█"
125 PRINT"Press C to continue"
126 GET U$:IF PEEK(17828)<>67 THEN 126
127 PRINT"█":R$:R$(R)
128 PRINTLI$
129 PRINT"T/S";TAB(10);"PARENT";TAB(25);"DAUGHTER"
130 PRINTLI$
131 FOR X=16TO 30
132 PRINTX;TAB(9);P(X);TAB(24);D(X)
133 NEXT
134 PRINTLI$
135 PRINT"T/S=timestep (timestep interval=NH/30)█"
136 REM
137 REM Display options available
138 REM
139 PRINT"A: ";LEFT$(R$(1-R),8);" E:end N:new data T:tabulate";
140 GET U$
141 IF U$="A" THEN PRINT"█":GOTO77
142 IF U$="E" THEN 146
143 IF U$="N" THEN 44
144 IF U$="T" THEN 116
145 GOTO 140
146 END
147 REM
148 REM Instructions
149 REM
150 PRINT"█"

```

```

151 PRINT"A radioactive substance (the 'parent')"
152 PRINT"decays in time to form a different,"
153 PRINT"'daughter' substance. The daughter may"
154 PRINT"be radioactive itself and decay in turn"
155 PRINT"to further 'daughters'."
156 PRINT"This program illustrates the decay"
157 PRINT"of the parent, and the build-up and"
158 PRINT"decay of its daughter."
159 PRINT"The number of atoms of parent and"
160 PRINT"daughter, relative to the original"
161 PRINT"number of parent atoms are shown as a"
162 PRINT"function of time."
163 PRINT"Parent and daughter ACTIVITY may also"
164 PRINT"be displayed (relative to the initial"
165 PRINT"activity of the parent). 'Activity' is"
166 PRINT"the rate at which radioactive atoms"
167 PRINT"disintegrate."
168 PRINT"Press C to continue"
169 GET U$:IF PEEK(17828)<>67 THEN 169
170 PRINT"@"
171 PRINT"The length of time for which the build"
172 PRINT"up and decays are calculated depends"
173 PRINT"on the number of parent half-lives you"
174 PRINT"enter when prompted to do so."
175 PRINT"You will also be prompted to enter the"
176 PRINT"value of the ratio: (parent half-life"
177 PRINT"divided by daughter half-life)."
178 PRINT"You may change from 'number of atoms'"
179 PRINT"to 'activity' by pressing key A when"
180 PRINT"prompted. Press T to obtain numerical"
181 PRINT"rather than graphical output. Press N"
182 PRINT"to enter new values of number of half-"
183 PRINT"lives to be displayed and parent to"
184 PRINT"daughter half-life ratio.Press E to end."
185 PRINT"@Press S to start."
186 GET U$:IF PEEK(17828)<>83 THEN 186
187 RETURN

```

Dear Sharpsoft,

I was pleased to see that you propose issuing your extremely informative S.U.N. six times yearly, but somewhat disappointed at the dearth of BASIC programmes. However, you have regularly made the point that "our" S.U.N. depends upon "our" contributions so I enclose two listings of programmes I have written, if you feel they would be of interest or use to our fellow members.

The Holiday Route Planner allows a route input from town to town (point to point) with an estimate of average "going" between the points (motorway, two lane carriageway, rural or urban) and within the programme (Lines 200-230) a means of modifying the speeds attributable to the "goings" input. A sample print is appended.

The programme of course is applicable to other areas which are sequential time dependent (e.g. production processes) and saves a considerable amount of pocket calculator work. Although written for DISC BASIC the program only requires LINES 1500-1630 to be amended for TAPE.

MEMBERS' LETTERS & LISTINGS

The second programme was written as a consequence of being requested to produce a series of friendly matches by the secretary of our golf club. He wanted, with a known number of members, to arrange a series of foursomes in which each member would play in the company of each other member and this program is the result.

I have found it very slow and would welcome suggestions as to how it might be speeded up as for 13 members (the second perfect solution) it takes about 25 minutes and I calculate that for the next perfect solution (82 players) it would take 8 weeks! (My definition of a perfect solution is where everyone plays in the company of everyone else just once and no non-pairing takes place).

This program also has applications in other areas - I have used it to calculate compound gear train alternatives - but then so do they all.

G.H. BENT

TOWN	Dist	¥	Route	Pa	Pa	Time	C/Dist	C/Time
1.START	0	T	0	0	0	0:00	0	0:00
2.POINT A	10	T	1	1	0:10	10	0:10	
3.POINT B	20	F	2	2	0:24	30	0:34	
4.POINT C	30	M	3	3	0:45	60	1:19	
5.POINT D	40	S	4	4	1:20	100	2:39	
6.FINISH	50	T	5	5	0:50	150	3:29	

T= 60 F= 50 M= 40 S= 30

```

100 REM-HOLIDAY ROUTE G.H.Bent 2 Jul 83.
110 DIM A$(200):I=1:LI$=""      TOWN      Dist ¥ Route Pa$":GOT0300
120 REM-T in mins. gives H hrs M mins
130 M=INT(T+.5):H=INT(M/60):M=M-60*H
140 H$=SPC(2-LEN(STR$(H)))+STR$(H):M$=STR$(M):IFM<10THENM$="0"+STR$(M)
150 TT$=H$+": "+M$:RETURN
160 T$=LEFT$(A$(I),20)
170 D$=MID$(A$(I),21,3)
180 G$=MID$(A$(I),24,1)
190 R$=MID$(A$(I),25,5)
200 P$=RIGHT$(A$(I),3)
210 N$=SPC(3-LEN(STR$(I)))+STR$(I):RETURN
220 IF(G$="T")THEN T=VAL(D$)/60+60:GOSUB130:Z1=60:RETURN
230 IF(G$="F")THEN T=VAL(D$)/50+60:GOSUB130:Z2=50:RETURN
240 IF(G$="M")THEN T=VAL(D$)/40+60:GOSUB130:Z3=40:RETURN
250 T=VAL(D$)/30+60:GOSUB130:Z4=30:RETURN
260 RETURN
270 PRINT:PRINT"      PRESS ANY KEY TO CONTINUE"
280 GETZ$:IFZ$=""THEN280
290 PRINTLI$:RETURN
300 PRINT"$$$":FORA=7T031:PRINTTAB(A):" ":":NEXTA:PRINT:PRINT
310 PRINT"

```


MEMBERS' LETTERS & LISTINGS

```

950 PRINT3,"AMEND/INSERT/DELETE###"
960 PRINT4,"PRINT ROUTE###"
970 PRINT5,"END      ###"
980 PRINT"  Enter option required ":
990 GETZ$:IFZ$=""THEN990
1000 Z=VAL(Z$):Z=ASC(Z$)-48:IF(Z<1)+(Z>5)THEN990
1010 PRINTZ:ONZGOTO1630,1030,1040,820,1020
1020 PRINT"0":END
1030 GOSUB480:GOTO920
1040 PRINT"####AMEND (A),INSERT (I),DELETE (D) ? ":
1050 GETZ$:IFZ$=""THEN1050
1060 IF(Z$="A")+(Z$="I")+(Z$="D")THEN1060
1070 GOTO1050
1080 PRINTZ$:GOSUB750
1090 IFZ$="A"THEN1120
1100 IFZ$="I"THEN1350
1110 IFZ$="D"THEN1440
1120 INPUT"0"  KEY ITEM NUMBER TO BE AMENDED ":I:PRINT
1130 GOSUB160
1140 PRINT"0"WHICH ELEMENT ? 0"
1150 PRINT1,"TOWN:";T$:PRINT"0";2,"MILES FROM LAST:";D$
1160 PRINT"0";3,"GOING:";G$:PRINT"0";4,"ROUTE No.:";R$
1170 PRINT"0";5,"PAGE No.:";P$:PRINT"0";6,"AMENDMENT COMPLETE"
1180 PRINT"0"  Key option ":
1190 GETZ$:IFZ$=""THEN1190
1200 Z=ASC(Z$):Z=Z-48:IF(Z<1)+(Z>6)THEN1190
1210 PRINTZ:"0":GOSUB160:ONZGOTO1220,1240,1260,1290,1310,1340
1220 PRINTT$:INPUT"NEW TOWN ? ":NT$:IFLEN(NT$)>20THENPRINT"00":GOTO1220
1230 T$=NT$+SPC(20-LEN(NT$)):GOTO1330
1240 PRINTD$:INPUT"NEW DISTANCE ? ":ND$:IFLEN(ND$)>3THENPRINT"000":GOTO1240
1250 D$=ND$+SPC(3-LEN(ND$)):GOTO1330
1260 PRINTG$:INPUT"NEW GOING ('T/F/M/S) ? ":NG$
1270 IF(NG$="T")+(NG$="F")+(NG$="M")+(NG$="S")THENG$=NG$:GOTO1330
1280 PRINT"000":GOTO1260
1290 PRINTR$:INPUT"NEW ROUTE No. ? ":NR$:IFLEN(NR$)>5THENPRINT"000":GOTO1290
1300 R$=NR$+SPC(5-LEN(NR$)):GOTO1330
1310 PRINTP$:INPUT"NEW PAGE No. ? ":NP$:IFLEN(NP$)>3THENPRINT"000":GOTO1310
1320 P$=NP$+SPC(3-LEN(NP$)):GOTO1330
1330 A$(I)=T$+D$+G$+R$+P$:GOTO1140
1340 GOSUB670:GOTO920
1350 INPUT"0"  KEY ITEM NUMBER BEFORE INSERTION ":II:PRINT:N=N+1
1360 FORI=NT0II+1STEP-1
1370 A$(I+1)=A$(I)
1380 NEXTI
1390 A$(II+1)=SPC(40)
1400 GOSUB490
1410 I=II+1:GOSUB530
1420 GOSUB670
1430 GOTO920
1440 INPUT"0"  KEY ITEM NUMBER TO BE DELETED ":II:I=II:PRINT
1450 GOSUB160
1460 PRINT"ITEM NUMBER ";N$
1470 PRINT"TOWN ";T$
1480 PRINT"DISTANCE ";D$
1490 PRINT"GOING ";G$
1500 PRINT"ROUTE NUMBER ";R$
1510 PRINT"PAGE NUMBER ";P$
1520 PRINT"IS THIS CORRECT ?"
1530 GETZ$:IFZ$=""THEN1530
1540 IFZ$="Y"THEN1570
1550 IFZ$="N"THEN1440
1560 GOTO1530
1570 FORI=IIT0N
1580 A$(I)=A$(I+1)
1590 NEXTI
1600 N=N-1

```

```

1610 GOSUB670
1620 GOTO920
1630 REM-CREATE ROUTE LIST
1640 INPUT"HOW MANY TOWNS/POINTS ON ROUTE ";N
1650 IFN>200THEN1640
1660 FORI=1TON
1670 GOSUB530
1680 NEXTI
1690 GOSUB680
1700 GOTO920

```

MEMBERS' LETTERS & LISTINGS

```

100 INPUT"NUMBER OF PLAYERS (MAX 60) ? ";N
110 DIMIA$(N),IB$(N),IC$(N),ID$(N),ZZ$(255)
120 GOTO810
130 REM-CHECK FOR INVALID COMBINATIONS
140 IF(IA$(A)="")+ (IB$(B)="")+ (IC$(C)="")+ (ID$(D)="") THENPP=1:RETURN
150 IF( (IB$(B)=IC$(C))+(IB$(B)=ID$(D)) THENPP=1:RETURN
160 IF( (IC$(C)=ID$(D)) THENPP=1:RETURN
170 RETURN
180 REM-CHECK FOR PREVIOUS COMBINATIONS
190 FORF=1TOE: IFPP=1THEN620
200 AB#=LEFT$(ZZ$(F),2)
210 IF( (AB#)=(IA$(A)+IB$(B)) THENPP=1:GOTO620
220 IF( (AB#)=(IB$(B)+IC$(C)) THENPP=1:GOTO620
230 IF( (AB#)=(IA$(A)+IC$(C)) THENPP=1:GOTO620
240 IF( (AB#)=(IB$(B)+ID$(D)) THENPP=1:GOTO620
250 IF( (AB#)=(IA$(A)+ID$(D)) THENPP=1:GOTO620
260 IF( (AB#)=(IC$(C)+ID$(D)) THENPP=1:GOTO620
270 AC#=LEFT$(ZZ$(F),1)+MID$(ZZ$(F),3,1)
280 IF( (AC#)=(IA$(A)+IB$(B)) THENPP=1:GOTO620
290 IF( (AC#)=(IB$(B)+IC$(C)) THENPP=1:GOTO620
300 IF( (AC#)=(IA$(A)+IC$(C)) THENPP=1:GOTO620
310 IF( (AC#)=(IB$(B)+ID$(D)) THENPP=1:GOTO620
320 IF( (AC#)=(IA$(A)+ID$(D)) THENPP=1:GOTO620
330 IF( (AC#)=(IC$(C)+ID$(D)) THENPP=1:GOTO620
340 AD#=LEFT$(ZZ$(F),1)+RIGHT$(ZZ$(F),1)
350 IF( (AD#)=(IA$(A)+IB$(B)) THENPP=1:GOTO620
360 IF( (AD#)=(IB$(B)+IC$(C)) THENPP=1:GOTO620
370 IF( (AD#)=(IA$(A)+IC$(C)) THENPP=1:GOTO620
380 IF( (AD#)=(IB$(B)+ID$(D)) THENPP=1:GOTO620
390 IF( (AD#)=(IA$(A)+ID$(D)) THENPP=1:GOTO620
400 IF( (AD#)=(IC$(C)+ID$(D)) THENPP=1:GOTO620
410 BC#=MID$(ZZ$(F),2,2)
420 IF( (BC#)=(IA$(A)+IB$(B)) THENPP=1:GOTO620
430 IF( (BC#)=(IB$(B)+IC$(C)) THENPP=1:GOTO620
440 IF( (BC#)=(IA$(A)+IC$(C)) THENPP=1:GOTO620
450 IF( (BC#)=(IB$(B)+ID$(D)) THENPP=1:GOTO620
460 IF( (BC#)=(IA$(A)+ID$(D)) THENPP=1:GOTO620
470 IF( (BC#)=(IC$(C)+ID$(D)) THENPP=1:GOTO620
480 BD#=MID$(ZZ$(F),2,1)+RIGHT$(ZZ$(F),1)
490 IF( (BD#)=(IA$(A)+IB$(B)) THENPP=1:GOTO620
500 IF( (BD#)=(IB$(B)+IC$(C)) THENPP=1:GOTO620
510 IF( (BD#)=(IA$(A)+IC$(C)) THENPP=1:GOTO620
520 IF( (BD#)=(IB$(B)+ID$(D)) THENPP=1:GOTO620
530 IF( (BD#)=(IA$(A)+ID$(D)) THENPP=1:GOTO620
540 IF( (BD#)=(IC$(C)+ID$(D)) THENPP=1:GOTO620
550 CD#=RIGHT$(ZZ$(F),2)
560 IF( (CD#)=(IA$(A)+IB$(B)) THENPP=1:GOTO620
570 IF( (CD#)=(IB$(B)+IC$(C)) THENPP=1:GOTO620
580 IF( (CD#)=(IA$(A)+IC$(C)) THENPP=1:GOTO620

```

MEMBERS' LETTERS & LISTINGS

```

590 IF((CD#)=(IB#(B)+ID#(D)))THENPP=1:GOTO620
600 IF((CD#)=(IA#(A)+ID#(D)))THENPP=1:GOTO620
610 IF((CD#)=(IC#(C)+ID#(D)))THENPP=1:GOTO620
620 NEXTF
630 RETURN
640 PRINT:PRINT"NO MATCHES FOR THE FOLLOWING"
650 FOR I = 1 TO N
660 PRINT IA#(I);" & ";
670 FORJ=ITON-1:Q=0
680 FORF=1TOE
690 A#=LEFT$(ZZ$(F),1):B#=MID$(ZZ$(F),2,1):C#=MID$(ZZ$(F),3,1)
700 D#=RIGHT$(ZZ$(F),1)
710 AB#=A#+B#:AC#=A#+C#:AD#=A#+D#:BC#=B#+C#:BD#=B#+D#:CD#=C#+D#
720 Z#=IA#(I)+IB#(J)
730 IF(Z#=AB#)+(Z#=AC#)+(Z#=AD#)+(Z#=BC#)+(Z#=BD#)+(Z#=CD#)THENQ=1
740 NEXTF
750 IFQ=1THENQ=0:GOTO770
760 PRINTIB#(J):CHR$(44)
770 NEXTJ
780 PRINT
790 NEXTI
800 RETURN
810 FORI=1TON:READIA#(I):NEXT:RESTORE
820 FORI=1TON-1:IB#(I)=IA#(I+1):NEXT
830 FORI=1TON-2:IC#(I)=IA#(I+2):NEXT
840 FORI=1TON-3:ID#(I)=IA#(I+3):NEXT
850 FORA=1TON
860 FORB=ATON-1
870 FORC=BTON-2
880 FORD=CTON-3
890 E#=IA#(A)+IB#(B)+IC#(C)+ID#(D)
900 PP=0:GOSUB130:IFPP=1THEN930
910 GOSUB180:IFPP=1THEN930
920 E=E+1:ZZ$(E)=E#:PRINTZZ$(E):PP=1
930 NEXTD
940 NEXTC
950 NEXTB
960 NEXTA
970 GOSUB640:END
980 DATA A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,a,b,c,d,e,f,g,h
990 DATA i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,ô,â,ä,ö,û,ü,ß,",@

```

Dear Sirs,

Wordpower and Epson FX-80
 CCCCCCCCCCCCCCCCCCCCCC

Can you please advise as to how I can achieve underlining, as the printer will print a line of CCCCC for the line written in as part of the text. (see example above)

It is possible to modify my Wordpower/Datapower packages so that they print out correctly on an Epson FX-80. When I purchased the two packages I was using an Epson MX-80, and the packages worked OK on that printer. The tapes do not appear to be listable to enable me to modify the programme to give underlining and it seemed reasonable to ask you what has to be done to make them compatible with the FX-80.

P.E. MORRISON
 KENT

The following will also apply to MZ-80A users.

Wordpower cannot output any control codes to enable underlining. The reason a row of C's appear is because Wordpower is passing the graphics characters of the Sharp MZ-80K (which you have obviously used to show an underline on the screen) to the FX-80. The FX-80 does not contain Sharp graphics in its character set, hence cannot print them and replaces them for another character, in your particular case a 'C'.

Your MX-80 must have had Sharp Graphics chips fitted to work correctly, using your method of typing a straight line on the screen underneath the word requiring underline.

The graphics chips are no longer available and have not been produced for the FX-80. Unfortunately due to the complexity of Wordpower it is unlikely that an underline feature will be incorporated.

SHARPSOFT

PLEASE DON'T FORGET,

WE ARE ALWAYS LOOKING FOR TIPS, HINTS, PEEKS, POKES AND PROBLEMS TO ANSWER AS WELL AS ANSWERS TO PROBLEMS. AUTHORS OF REALLY INTERESTING ARTICLES OR LISTINGS WILL RECEIVE A FIVE POUND SHARPSOFT VOUCHER TO SHOW OUR APPRECIATION.

WE WILL ALSO BE PLEASED TO EVALUATE ANY SOFTWARE WITH A VIEW TO INCLUSION ON OUR SOFTWARE LIST. SO IF YOU HAVE A "PRIDE & JOY" OR HAVE BEEN THINKING ABOUT ONE! - SEND IT IN ON TAPE OR DISK WITH AS MUCH DETAIL (DESCRIPTION, INSTRUCTIONS, LISTING ETC.) AS POSSIBLE AND WE WILL CERTAINLY LOOK VERY CAREFULLY AT IT. HOWEVER, PLEASE DO NOT SEND US YOUR ONLY COPY OF THE DISK OR TAPE AS WE CANNOT BE RESPONSIBLE FOR ANY LOSS OR DAMAGE EN ROUTE TO US.

SHARPSOFT USER NOTES
SHARPSOFT LIMITED
CRISALLEN HOUSE
2nd FLOOR
86/90 PAUL STREET
LONDON, EC2A 4NE

TEL: 01-739-8559

SHARPSOFT

Sharpsoft Ltd., 86-90 Paul Street, London EC2A 4NE

Printed by Oldham Press (T.U.), Chatham, Kent.