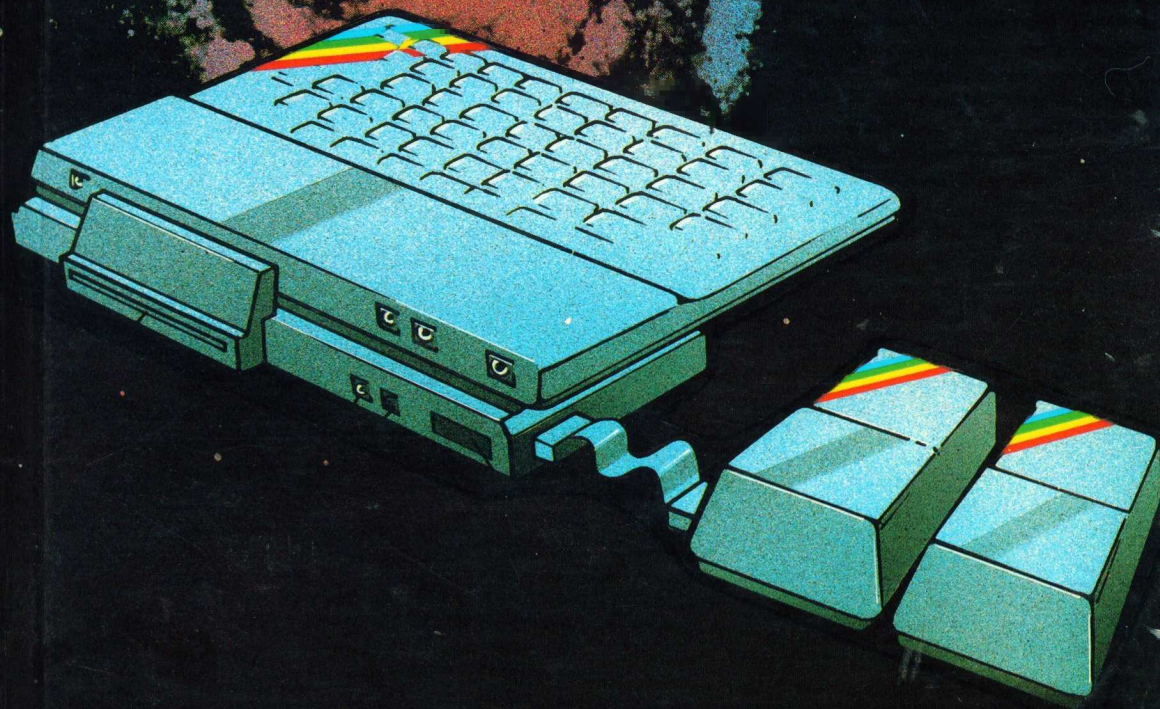


# MAKING THE MOST OF YOUR SPECTRUM MICRO DRIVES

Richard G. Hurley



# **Making the Most of your Spectrum Micro Drives**

Richard G. Hurley



**MICRO PRESS**

First published in 1984 in the United Kingdom by  
Micro Press  
Castle House, 27 London Road  
Tunbridge Wells, Kent

© Richard G. Hurley 1984

All rights reserved. No part of  
this publication may be reproduced,  
stored in a retrieval system, or  
transmitted in any form or by any  
means, electronic, mechanical, recording  
or otherwise, without the prior  
permission of the publishers.

British Library Cataloguing in Publication Data

Hurley, Richard G.

Making the most of your Spectrum Micro Drives.

1. Sinclair ZX Spectrum (Computer)

2. Micro Drives

I. Title

001.64'42      QA76.8.S625

ISBN 0-7447-0005-1

Typeset by Keyset Composition, Colchester  
Printed and bound by Mackays of Chatham Ltd

To JOSIE  
for all her active encouragement

# Contents

<i>Introduction</i>	ix	
<i>Chapter 1</i>	<b>The Micro Drives</b>	1
<i>Chapter 2</i>	<b>Channels and Streams</b>	6
<i>Chapter 3</i>	<b>Files and Data Processing</b>	10
<i>Chapter 4</i>	<b>The Address Book</b>	23
<i>Chapter 5</i>	<b>The Electronic Diary</b>	38
<i>Chapter 6</i>	<b>Large Data Files</b>	64
<i>Chapter 7</i>	<b>English/French Translator</b>	71
<i>Chapter 8</i>	<b>The Network</b>	80
<i>Chapter 9</i>	<b>The RS232 Interface</b>	97
<i>Chapter 10</i>	<b>Slide Show</b>	105
<i>Chapter 11</i>	<b>The ZX-Interface.2</b>	113
<i>Index</i>		151

## Acknowledgements

I should like to express my appreciation of my students at Hurstpierpoint College who, over the past few months, have helped to refine the programs contained within this book. I am particularly grateful to James Mead and Steven Lacey for their active participation in the formulation of several programs. I would also like to thank Mr David Virgo and Mr Keith Weber, without whose support this book would never have been finished on time. Finally, I should like to thank my long-suffering wife, Josie, and two daughters Melissa and Georgina, without whose encouragement and patience this book would never have been completed.

Richard G. Hurley  
Hurstpierpoint, November 1983.

# Introduction

The ZX-Interface.1 and micro drives increase the power of the Spectrum computer many times over. The interface itself not only acts as the drive controller but also facilitates communication with peripheral devices such as printers, plotters, etc., and directly communicates with other Spectrum computers.

This book does not set out with the intention of becoming a technical manual for these devices, nor does it assume a wide knowledge of computing beyond that covered in the Spectrum manual. It is aimed at showing the user, with the aid of suitable examples, how to get the best out of his machine, and how it can be used for data processing, peripheral control and networking.

The increased speed and capacity, obtained through the use of micro drives, means that quite advanced file handling can be achieved without having to resort to the complexities of machine code. It is because of this that all the programs contained within this book are written in BASIC only.

As well as being instructive, the book also contains several useful long programs that I have found invaluable both at home and at work. These programs are fully documented and it is hoped that the user will learn from the procedures used.

# The Micro Drives

In this chapter we shall be considering the extensions to BASIC made available by the user of the ZX-Interface.1 in conjunction with one or more micro drive units.

Fig. 1 shows all the new commands available for use with the drives.

```
FORMAT "m";n;"NAME"  
CAT n  
ERASE "m";n;"NAME"  
SAVE* "m";n;"NAME"  
LOAD* "m";n;"NAME"  
VERIFY* "m";n;"NAME"  
MERGE* "m";n;"NAME"
```

Fig. 1.

The notation used in the above commands is as follows:

- "m" Small or capital m signifying communication with a micro drive;
- n A number (in the range 1-8) indicating which drive is being used;
- "NAME" A string indicating the name of the program, data file or cartridge.

## Commands

Let us now look at each of the above terms in more detail, considering the method of entry, their use and any variations that can be employed.



### **FORMAT "m";n;"NAME"**

Before a micro drive cartridge can be used, it must be formatted. This will take about 30 seconds, during which time the computer will check the tape and assign the cartridge a name with up to ten characters.

The job of formatting a new cartridge can be achieved from within a program, as shown in following example and as used in the diary program found in chapter 5.

```
9000 REM FORMAT ROUTINE
9010 INPUT "DRIVE NUMBER=";n
9020 IF n>8 THEN GO TO 9010
9030 INPUT "CARTRIDGE NAME=";A$
9040 IF LEN(A$)>10 THEN GO TO 9030
9050 PRINT TAB(12);"FORMATTING"
9060 FORMAT "m";n;A$
9070 STOP
```

#### *Note*

Observe that it is possible to use variables within the FORMAT statement. This is satisfactory provided they have been assigned values using LET, READ or INPUT statements.

### **CAT n**

The CAT or CATALOG command is used to display the contents of a micro drive cartridge on the screen. The type of display is shown in Fig. 2.

TEST	(Cartridge name)
ALPHA	
BETA	(Program names in alphabetical order)
GAMMA	
46	(Number of kilobytes left)

*Fig. 2.*

### **ERASE "m";n;"NAME"**

This command is used to remove a program or data file from a micro drive cartridge. We are not able to overwrite a program

with another program of the same name, as is possible using tapes. Consequently, the original version must first be removed using the ERASE command.

Consider the problem of having a program called TEST which is to be replaced on a cartridge by an updated version with the same name. To achieve this the following three commands would be necessary

```
ERASE      "m";1;"TEST"
SAVE*     "m";1;"TEST"
VERIFY*   "m";1;"TEST"
```

#### Note

As with most of the Spectrum commands, ERASE can be used within a BASIC program and it can also make use of variables to represent the drive number and program name. These facts are of great importance in data processing and are demonstrated in many of the programs found in this book.

#### SAVE\* "m";n;"NAME"

The SAVE\* command is similar in operation to the SAVE command used with the basic, unexpanded Spectrum, except that the operation of dumping the program onto the storage medium is many times faster.

The syntax of the command is slightly increased by the inclusion of the channel specifier (m) and the drive number (n), but in all other ways the rules governing the syntax are the same. We are therefore able to save programs, arrays, machine code and screen\$ using the format

```
Program      SAVE* "m";n;"NAME"
Array        SAVE* "m";n;"NAME" DATA A$( )
Machine code SAVE* "m";n;"NAME" CODE A,B
Screen$      SAVE* "m";n;"NAME" SCREEN$
```

#### Note

As with the SAVE command, the micro drive SAVE\* command can also be used in conjunction with the LINE statement to enable the program to begin at a particular line as soon as it has loaded. For example

```
SAVE* "m";n;"NAME" LINE 100
```

The program would then be automatically executed from line 100 after being loaded from the cartridge.

### **Auto-run Facility**

When using the ZX-Interface.1 and a micro drive, there is a built-in facility which will allow a program to be run automatically from a cartridge without having to enter the long LOAD\* command. This is achieved by finishing the program with the line

```
9999 SAVE* "m";1;"run" LINE 10
```

and then saving it by typing

```
GOTO 9999
```

After the program has been transferred to the cartridge, it can be loaded and run automatically by entering NEW followed by RUN. *This will only work on drive.1, and the program must be called "run" (n.b. not capitals).*

### **VERIFY\* "m";n;"NAME"**

The micro drive VERIFY\* command is very similar to the VERIFY command but includes the extra information required for communication with the drives.

As with the cassette-operating system, programs, arrays and machine code can be checked, but we also have the additional facility that a SCREEN\$ can be verified. The syntax for these operations is:

```
VERIFY* "m";n;"NAME"  
VERIFY* "m";n;"NAME" DATA A$( )  
VERIFY* "m";n;"NAME" CODE  
VERIFY* "m";n;"NAME" SCREEN$
```

### **LOAD\* "m";n;"NAME"**

If we take into account the extended syntax required for micro drive operation, then there is no difference between the LOAD command and LOAD\*.

**MERGE \* "m";n;"NAME"**

The MERGE command is very similar to the LOAD command, except that any program lines currently in memory will not be lost. This facility is very useful when a program is to be extended, since this can be achieved during the execution of the program.

```

9000 REM *COLLECT NEW ROUTINE *
9010 INPUT "DO YOU REQUIRE AN EXTRA ROUTINE";A$
9020 IF A$="NO" THEN RETURN
9030 INPUT "NAME OF ROUTINE="";A$
9040 INPUT "DRIVE NUMBER="";A
9050 MERGE * "m";A;A$

```

*Note*

The MERGE command will not work with any program saved using the SAVE\* . . . LINE command.

# Channels and Streams

At first glance, the concept of a stream or a channel is quite complex. It is useful, therefore, to have an understanding of the structure of a computer system, and from this the necessity of such terminology becomes obvious.

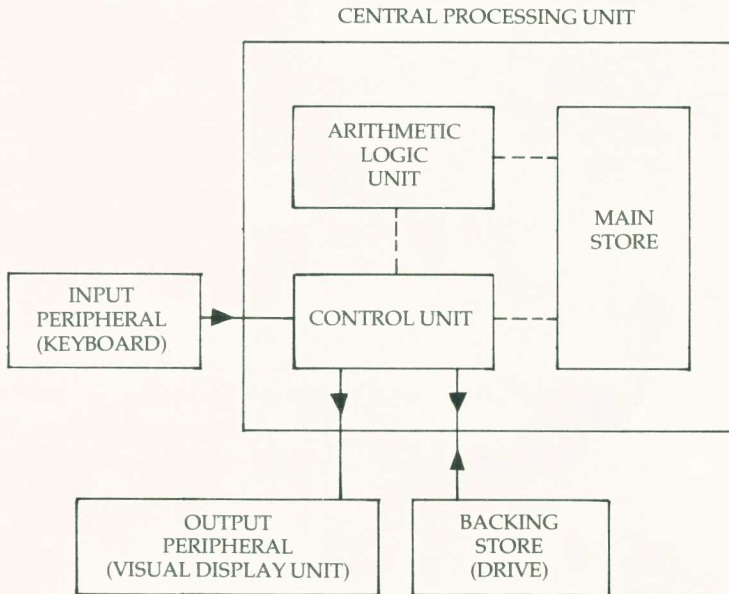


Fig. 3.

Fig. 3 represents a typical computer system with the central processing unit (CPU) communicating with the various peripheral devices via the control unit. In the Spectrum, as in the case of many larger computers, the peripherals with which the CPU communicates are called *channels* and the routes along which this communication takes place are the *streams*.

## Channels

Table 1 shows the channels accessible to the user and indicates their nature (input/output) and the specifier used in the relevant commands.

It should be observed that the RS232 interface has two distinct channels — b for binary-coded information and t for text — and these will be considered in greater depth later (see Chapter 9).

**Table 1**

Channel	Type	Specifier
Screen	Output	s
Keyboard	Input	k
ZX printer	Output	p
Micro drive	Input/output	m
Network	Input/output	n
RS232	Input/output	b or t

## Streams

As described above, the routes by which the CPU communicates with the peripherals (channels) are termed streams. The Spectrum has 16 of these streams (numbered 0–15) available; of these the first four are permanently linked to the most used channels. These important links are shown in Table 2 and the first three are best left and should not be 'opened' or 'closed'.

**Table 2**

Stream	Channel
#0	Bottom of screen/keyboard
#1	
#2	Upper part of screen
#3	ZX printer

## Commands

There are three special commands that deal with the channels and streams of the Spectrum. These are OPEN#, CLOSE# and MOVE. The format for each is examined below and further applications can be seen in most of the programs found in subsequent chapters.

### OPEN#

The OPEN# command is used to connect a particular channel to the CPU using a stream of one's choice. Once opened communication can continue along this stream to the channel until a CLOSE# command is encountered. Some examples of how the OPEN# command can be used are:

- 1 . . . OPEN#4;"m";1;"TEST"
- 2 . . . OPEN#4;"t"
- 3 . . . OPEN#5;"b"
- 4 . . . OPEN#6;"n";2

### COMMENTS

- Line 1 Opens a stream to micro drive.1 along which a data file called TEST will be sent.
- Line 2 Opens a stream to the RS232 interface for the transmission of text.
- Line 3 Opens a stream to the RS232 interface for the transmission of binary codes.
- Line 4 Opens a stream to station.2 on the network system.

### CLOSE#

After a stream has been opened and all communication with the device has been completed, the stream must be closed.

### MOVE

The MOVE command is very useful because it enables information to be transferred directly from one channel or stream to another, that is the command MOVE#1 to #3 will transfer data from the keyboard directly to the printer. By careful use of the

MOVE command some very useful short cuts can be implemented, as shown in the following examples.

*Example 1*

```
MOVE "m";1;"TEST" TO #2
```

This will transfer the whole of the data file TEST from micro drive.1 onto the screen. This is very useful when checking if a particular data file has become corrupt.

*Example 2*

```
MOVE "m";1;"TEST" TO #3
```

This will dump the data file TEST onto the ZX-printer.

*Example 3*

```
MOVE "m";1;"TEST" TO "m";2;"TEST"
```

This will make a back-up copy on drive.2 of the file TEST, originally on drive.1.



# Files and Data Processing

One of the major advantages of a computer is its ability to store large quantities of information, which it can then search, process and display at very high speed. The method of dealing with information in this way is generally referred to as *data processing* (DP). In this chapter, we will see how quite sophisticated work can be done using a standard Spectrum with the ZX-Interface.1 and a single micro drive unit.

## Files

In data processing, the quantity of information being considered is often very large. It is therefore impractical to store all the variables in data statements or in arrays within the program. Under such circumstances, the data would be stored in a file which could be extracted from the backing store under direct program control.

### File Structure

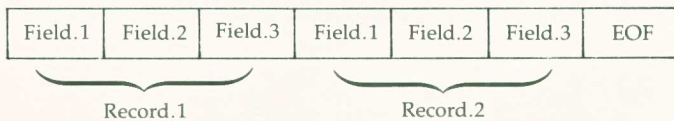


Fig. 4.

Fig. 4 shows how a file is divided up into a series of *records* with the individual records being divided into a number of *fields*. Each field can be numeric, alphabetic or alphanumeric and can contain one item of data (e.g., a name, number, wage, date, etc.).

When handling a file, it is imperative to remember its structure, paying special attention to:

- 1) The number of fields per record;
- 2) The type of each field;
- 3) Which field represents the *key*;
- 4) The type of *end of file marker* used.

### *Key field*

In general, the first field of each record is called the *key field*. This is used to identify a particular record, separating it from all the others in the file. In most cases, the key field will be a name, but this is not necessarily the case, since some type of code could be used to protect the integrity of the data.

### *End of file (EOF)*

When the computer is required to read the information contained within a file, it is most unlikely that it will have prior knowledge of the number of records which the file contains. If, under these circumstances, we attempt to read the file by using a *for-next* loop, then it is highly likely that the situation will arise where the computer is attempting to extract records from a file that contains no more data. To overcome this problem it is usual to end a file with an EOF marker. When this is encountered, the computer knows that there is no more data, and the reading procedure should now cease and the stream must be closed.

## Commands

There are four new commands which are part of the extended system which will be used throughout the next few chapters dealing with files and file management using the Spectrum. These commands are shown in Fig. 5.

<pre> OPEN#N;"m";n;"NAME" CLOSE#N PRINT#N; INPUT#N; </pre>
------------------------------------------------------------

Fig. 5.

**OPEN#N;"m";n;"NAME"**

This will open the stream N (in the range 4–15) from the CPU to the micro drive n along which a file called NAME is to be sent. All the values N,n and NAME could be replaced by suitable variables, provided they have been previously assigned.

**CLOSE#N**

After the transmission of a data file along a stream has been completed, the stream must be disconnected or closed by using the CLOSE# command.

**PRINT#N;**

The PRINT# command is used when writing a file to a micro drive. Its effect is to send one item of data or one field, which is then stored on the tape.

*Note*

According to the manual, we are able to have multiple PRINT# statements by making use of the apostrophe: for example

```
PRINT#4;A$'B$
```

However, in the programs found in this book, no use is made of such multiple lines and all printing to file is done with separate statements. The reason for this is that any mistakes (i.e. missed apostrophes) would result in corrupt and possibly unreadable files.

**INPUT#N;**

This command is the opposite of the PRINT# statement, transferring one item of data (field) from the file into the main store. Again, we are able to use multiple statements, using a semicolon as a separator. It is the policy of this book, however, to encourage only one input per line. This is unlikely to cause too much extra programming, but it could save a great many problems!

**Types of DP Programs**

There are three distinct types of DP or file-handling programs:

- 1) Write only;

- 2) Read only;
- 3) Read and write.

As the names suggest, the first type is used to construct an original file, the second to interrogate or read an existing file and the third, and final, type is able to read, change and then rewrite the file back onto tape.

### Write only

In this category of program, the object is to create a file on tape where none previously existed. To achieve this the data must first be collected using LET, READ or INPUT instructions, then checked and finally transferred to a file using the PRINT# statement.

#### Example 1

The following program constructs a file called PHONE, which contains a series of people's names and telephone numbers. Two string arrays — A\$( ) and B\$( ) — are used to hold the information in main store and an EOF is used to signify the end of the file.

```

10 CLS
20 DIM A$(50,12)
30 DIM B$(50,12)
40 PRINT TAB (10);"PHONE NUMBE
RS"
50 PRINT TAB (10);"===== =====
=="
60 PRINT "NAME","NUMBER"
70 PRINT "====","=====
80 LET I=1
90 INPUT "NAME=";A$(I)
100 IF A$(I,1 TO 3)="END" THEN
GO TO 150
110 INPUT "NUMBER=";B$(I)
120 PRINT A$(I),B$(I)
130 LET I=I+1
140 GO TO 90
150 PRINT : PRINT
160 PRINT INVERSE 1;"WRITING F
ILE"; INVERSE 0
170 OPEN #4;"M";1;"PHONE"

```

```

180 FOR J=1 TO I-1
190 PRINT #4;A$(J)
200 PRINT #4;B$(J)
210 NEXT J
220 PRINT #4;"EOF"
230 CLOSE #4
240 STOP

```

## COMMENTS

- Lines 10-70     Print titles and initialize the variables.  
 Lines 80-150    Input, via the keyboard, a series of names and telephone numbers.  
 Line 170        Open stream #4 to channel m along which a data file called PHONE is to be transferred.  
 Lines 180-210   Print all the names and numbers onto the file.  
 Line 220        Send EOF marker to the file.  
 Lines 230-240   Close file and end program.

When this program has finished running a file called PHONE with a structure as indicated in Fig. 6 will exist on tape.

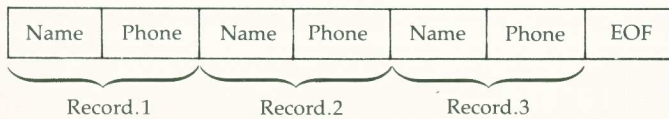


Fig. 6.

**Read only**

Having created a file on tape using a write only program, as demonstrated in Example 1, there will come a time when it is necessary to interrogate the file in order to extract a piece, or pieces, of information. If we only need to obtain or process the data from the file, then a program such as that shown in Example 2 would be sufficient.

*Example 2*

The following program reads the telephone file created in Example 1, and then gives the user the choice of:

- 1) Printing a table of numbers;
- 2) Searching for a name and displaying the corresponding number.

```

10 CLS
20 DIM A$(50,12)
30 DIM B$(50,12)
40 PRINT TAB (10);"TELEPHONE P
ROGRAM"
50 PRINT TAB (10);"===== =
===== "
60 PRINT : PRINT
70 PRINT INVERSE 1;"FILE LOAD
ING"; INVERSE 0
80 OPEN #4;"M";1;"PHONE"
90 LET I=1: LET TEST=0
100 INPUT #4;A$(I)
110 IF A$(I,1 TO 3)="EOF" THEN
GO TO 150
120 INPUT #4;B$(I)
130 LET I=I+1
140 GO TO 100
150 CLOSE #4
160 CLS
170 PRINT TAB (10);"MAIN MENU"
180 PRINT TAB (10);"==== ====="
190 PRINT : PRINT : PRINT
200 PRINT "1....PRINT TABLE OF
NUMBERS"
210 PRINT : PRINT
220 PRINT "2....SEARCH AND DISP
LAY"
230 PRINT : PRINT
240 PRINT "3....EXIT"
250 INPUT "SELECTION=";A
260 IF A<1 OR A>3 THEN GO TO 2
50
270 GO TO (A*1000)
1000 REM *TABLE*
1010 CLS
1020 PRINT TAB (10);"TELEPHONE N
UMBERS"
1030 PRINT TAB (10);"===== =
===== "
1040 PRINT
1050 PRINT "NAME","NUMBER"
1060 PRINT "====","====="
1070 FOR J=1 TO I-1

```

```

1080 PRINT A$(J),B$(J)
1090 NEXT J
1100 INPUT "PRESS ENTER TO CONTI
NUE.";Z$
1110 GO TO 160
2000 REM *SEARCH & DISPLAY*
2010 CLS
2020 PRINT TAB (8);"SEARCH AND D
ISPLAY"
2030 PRINT TAB (8);"===== === =
===== "
2040 PRINT
2050 INPUT "NAME=";N$
2060 FOR J=1 TO I-1
2070 IF A$(J,1 TO LEN (N$))=N$ T
HEN PRINT "NAME=";A$(J): PRINT
"NUMBER=";B$(J): LET TEST=TEST+1
: PRINT
2080 NEXT J
2090 IF TEST<>0 THEN GO TO 2120
2100 PRINT : PRINT : PRINT
2110 PRINT INVERSE 1;"NO SUCH R
ECORD ON FILE"; INVERSE 0
2120 INPUT "PRESS ENTER TO CONTI
NUE.";Z$
2130 GO TO 160
3000 REM *EXIT*
3010 STOP

```

## COMMENTS

- |                 |                                                                                                                                           |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| Lines 10-60     | Print titles and initialize the variables.                                                                                                |
| Lines 70-150    | Input the file PHONE from a cartridge in micro drive.1.                                                                                   |
| Lines 160-270   | Main menu giving the user a choice of three options.                                                                                      |
| Lines 1000-1110 | Computer displays all the names and numbers contained within the PHONE file.                                                              |
| Lines 2000-2130 | User inputs a name, via the keyboard, and the computer searches the data file for the appropriate information, printing it on the screen. |
| Line 3000       | Exit.                                                                                                                                     |

**Read/Write**

The third and most useful type of file-handling program is a read/write or *file management* program. As the title suggests, the program assumes that a file already exists, which can then be loaded into the computer, interrogated, changed and then written back onto the tape.

The number of facilities contained within such a program will depend on its sophistication, but will usually allow the user to:

- 1) Change a record;
- 2) Delete a record;
- 3) Add a record;
- 4) Interrogate the file.

*Example 3*

The following is a file management program for handling the PHONE data file created in Example 1 and includes all the facilities listed above.

```

10 CLS
20 DIM A$(50,12)
30 DIM B$(50,12)
40 DIM D(20)
50 PRINT TAB (10);"TELEPHONE M
ANAGER"
60 PRINT TAB (10);"===== =
===== "
70 PRINT : PRINT
80 PRINT INVERSE 1;"FILE LOAD
ING"; INVERSE 0
90 OPEN #4;"M";1;"PHONE"
100 LET I=1: LET TEST=0: LET CH
=0
110 INPUT #4;A$(I)
120 IF A$(I,1 TO 3)="EOF" THEN
GO TO 160
130 INPUT #4;B$(I)
140 LET I=I+1
150 GO TO 110
160 CLOSE #4
170 CLS

```



```

180 PRINT TAB (10);"MAIN MENU"
190 PRINT TAB (10);"==== ====="
200 PRINT : PRINT
210 PRINT "1....PRINT TABLE"
220 PRINT
230 PRINT "2....SEARCH AND DISP
LAY"
240 PRINT
250 PRINT "3....ADD A RECORD"
260 PRINT
270 PRINT "4....DELETE A RECORD
"
280 PRINT
290 PRINT "5....CHANGE A RECORD
"
300 PRINT
310 PRINT "6....EXIT PROGRAM"
320 INPUT "SELECTION=";A
330 IF A<1 OR A>6 THEN GO TO 3
20
340 GO TO (A*1000)
1000 REM *TABLE*
1010 CLS
1020 PRINT TAB (10);"TELEPHONE N
UMBERS"
1030 PRINT TAB (10);"===== =
===== "
1040 PRINT
1050 PRINT "NAME", "NUMBER"
1060 PRINT "====", "====="
1070 FOR J=1 TO I-1
1080 PRINT A$(J),B$(J)
1090 NEXT J
1100 INPUT "PRESS ENTER TO CONTI
NUE.";Z$
1110 GO TO 170
2000 REM *SEARCH AND DISPLAY*
2010 CLS
2020 PRINT TAB (8);"SEARCH AND D
ISPLAY"
2030 PRINT TAB (8);"===== === =
===== "
2040 PRINT
2050 INPUT "NAME=";N$

```

```

2060 FOR J=1 TO I-1
2070 IF A$(J,1 TO LEN (N$))=N$ T
HEN PRINT "NAME=";A$(J): PRINT
"NUMBER=";B$(J): LET TEST=TEST+1
2080 PRINT
2090 NEXT J
2100 IF TEST<>0 THEN GO TO 2130
2110 PRINT : PRINT : PRINT
2120 PRINT INVERSE 1;"NO SUCH R
ECORD ON FILE"; INVERSE 0
2130 INPUT "PRESS ENTER TO CONTI
NUE.";Z$
2140 LET TEST=0
2150 GO TO 170
3000 REM *NEW RECORDS*
3010 CLS
3020 PRINT TAB (10);"NEW RECORDS
"
3030 PRINT TAB (10);"=== =====
"
3040 PRINT : PRINT : PRINT
3050 INPUT "NAME=";A$(I)
3060 PRINT "NAME=";A$(I)
3070 PRINT
3080 INPUT "NUMBER=";B$(I)
3090 PRINT "NUMBER=";B$(I)
3100 LET I=I+1
3110 LET CH=CH+1
3120 INPUT "PRESS ENTER TO CONTI
NUE.";Z$
3130 GO TO 170
4000 REM *DELETE RECORD*
4010 CLS
4020 PRINT TAB (10);"DELETE RECO
RD"
4030 PRINT TAB (10);"===== =====
=="
4040 PRINT
4050 INPUT "NAME=";N$
4060 LET L=1
4070 FOR J=1 TO I-1
4090 IF A$(J,1 TO LEN (N$))=N$ T
HEN PRINT L;".";A$(J),B$(J): LE
T D(L)=J: LET L=L+1

```

```

4100 NEXT J
4110 IF L=1 THEN GO TO 4210
4120 INPUT "RECORD NO. TO BE DEL
ETED=";R
4130 FOR K=D(R) TO I-1
4140 LET A$(K)=A$(K+1)
4150 LET B$(K)=B$(K+1)
4160 NEXT K
4170 LET I=I-1
4180 PRINT : PRINT
4190 PRINT INVERSE 1;"RECORD DE
LETED"; INVERSE 0
4200 GO TO 4230
4210 PRINT : PRINT
4220 PRINT INVERSE 1;"NO SUCH R
ECORD"; INVERSE 0
4230 INPUT "PRESS ENTER TO CONTI
NUE.";Z$
4240 GO TO 170
5000 REM *CHANGE RECORD*
5010 CLS
5020 PRINT TAB (10);"CHANGE RECO
RD"
5030 PRINT TAB (10);"=====
====="
5040 PRINT
5050 INPUT "NAME=";N$
5060 LET L=1
5070 FOR J=1 TO I-1
5080 IF A$(J,1 TO LEN (N$))=N$ T
HEN PRINT L;".";A$(J),B$(J): LE
T D(L)=J: LET L=L+1
5090 NEXT J
5100 IF L=1 THEN GO TO 5150
5110 INPUT "RECORD NO. TO BE CHA
NGED=";R
5120 INPUT "NEW TELEPHONE NUMBER
=";B$(D(R))
5130 LET CH=CH+1
5140 GO TO 5170
5150 PRINT : PRINT
5160 PRINT INVERSE 1;"NO SUCH R
ECORD"; INVERSE 0
5170 INPUT "PRESS ENTER TO CONTI
NUE.";Z$

```

```

5180 GO TO 170
6000 REM *EXIT PROGRAM*
6010 IF CH=0 THEN STOP
6020 ERASE "M";1;"PHONE"
6030 OPEN #4;"M";1;"PHONE"
6040 FOR J=1 TO I-1
6050 PRINT #4;A$(J)
6060 PRINT #4;B$(J)
6070 NEXT J
6080 PRINT #4;"EOF"
6090 CLOSE #4
6100 STOP

```

#### COMMENTS

- Lines 10–70            Print titles and initialize the variables.
- Lines 80–160         Input the file PHONE from a cartridge in micro drive.1.
- Lines 170–340        Main menu giving the user the choice of six different options.
- Lines 1000–1110      Computer displays all the names and numbers contained within the file PHONE.
- Lines 2000–2150     User inputs a name and the computer searches the data file for the appropriate information, printing it on the screen.
- Lines 3000–3130     User is able to input a new record (name and number), which the computer inserts at the end of the array.
- Lines 4000–4240     User is able to delete a complete record. This is achieved by finding the required record and then moving all further records along the array, thus overwriting the one to be deleted.
- Lines 5000–5180     User is able to change a record; that is, the telephone number field is altered but the key name field remains the same.
- Lines 6000–6100     If any alterations have been made (CH<>0), then the file PHONE is first erased from tape and a new updated version is then recorded.

#### Variables

Table 3 shown below lists the main variables used in Examples 1, 2 and 3. This may be of use in appreciating how the programs operate.

**Table 3**

---

Name	Use
A\$( )	Names
B\$( )	Numbers
I	Number of records
CH	Variable counting the number of alterations made
TEST	Variable set to 1 when a suitable record is located
D(L)	An array used to list all records found in change/delete sections
L	Number of records in change/delete sections
R	Record number to be changed/deleted

---

## Conclusion

The techniques discussed in this chapter and introduced in Examples 1, 2 and 3 will be put to practical use in all the programs found in the next two chapters.

## Chapter 4

# The Address Book

This chapter contains two programs (ADDRESS.1 and ADDRESS.2) which can be typed into the computer and then saved on a micro drive cartridge. When both programs have been entered and recorded, they will provide the user with an electronic 'address book' capable of storing and handling over 100 names and addresses.

The system will operate only if both programs are saved on tape and then used in accordance with the instructions found at the end of this chapter and within the programs themselves. They are designed to be fool-proof and self-instructing, so that once entered they can be put to good use by any member of the family without having to refer to this book.

## ADDRESS.1

This is the write only program which initializes the address book. It should be typed into the computer and then saved using the command

```
SAVE *"m";1;"ADDRESS.1"
```

### *Listing*

```
10 LET I=1
20 DIM t$(50,30)
30 DIM u$(50,30)
40 DIM v$(50,30)
50 DIM w$(50,30)
60 DIM x$(50,30)
70 DIM y$(50,30)
80 DIM z$(50,30)
```

```

90 CLS
100 PRINT TAB (9);"FILE MAKER"
110 PRINT TAB (9);"==== ====="
120 PRINT : PRINT
130 PRINT "THIS PROGRAM IS USED
TO SET UP"
140 PRINT
150 PRINT "A DATA FILE CALLED A
DDRESS"
160 PRINT
170 PRINT "ON A MICRO DRIVE, WH
ICH IF USED"
180 PRINT
190 PRINT "IN CONJUNCTION WITH
THE FILE"
200 PRINT
210 PRINT "MANAGEMENT PROGRAM..
. . ."
220 PRINT
230 PRINT "                ADDRESS
.2"
240 PRINT
250 PRINT "WILL GIVE THE FULL F
ACILITIES OF"
260 PRINT
270 PRINT "AN ELECTRONIC ADDRE
SS BOOK."
280 INPUT "PRESS ENTER TO CONTI
NUE.";A$
290 CLS
300 PRINT "THE INITIALIZATION P
ROCESS"
310 PRINT
320 PRINT "WILL INVOLVE THE ENT
ERING OF"
330 PRINT
340 PRINT "A SERIES OF NAMES AN
D ADDRESSES."
350 PRINT
360 PRINT "(THIS CAN BE ANY NO.
FROM 1-20.)"
370 PRINT
380 PRINT "WHEN THE LAST NAME A
ND ADDRESS"

```

```

390 PRINT
400 PRINT "HAS BEEN ENTERED, TH
EN ENTERING"
410 PRINT
420 PRINT "THE NAME 'END' WILL
SAVE THE"
430 PRINT
440 PRINT "FILE."
450 INPUT "PRESS ENTER TO CONTI
NUE.";A$
460 CLS
470 PRINT "ONE PROBLEM WHICH MA
Y OCCUR IS "
480 PRINT
490 PRINT "THAT DIFFERENT ADDRE
SSES WILL BE"
500 PRINT
510 PRINT "OF DIFFERENT LENGTHS
"
520 PRINT
530 PRINT "THIS IS OVERCOME BY
SETTING ALL"
540 PRINT
550 PRINT "ENTRIES TO SEVEN LIN
ES WITH THE"
560 PRINT
570 PRINT "ENTER KEY BEING USED
IF NOTHING"
580 PRINT
590 PRINT "IS REQUIRED."
600 INPUT "PRESS ENTER TO CONTI
NUE.";A$
610 CLS
620 PRINT "EXAMPLE"
630 PRINT "====="
640 PRINT : PRINT
650 PRINT "NAME.....MR.R.G.H
URLEY"
660 PRINT "LINE(2).....26 ST.JO
HNS LANE"
670 PRINT "LINE(3).....WORTHING
"
680 PRINT "LINE(4).....SUSSEX"
690 PRINT "LINE(5).....BN2 2JK"

```



```

700 PRINT "LINE(6)....."; INK 2
; FLASH 1;"(PRESS ENTER)"; INK 0
; FLASH 0
710 PRINT "LINE(7)....."; INK 2
; FLASH 1;"(PRESS ENTER)"; INK 0
; FLASH 0
720 INPUT "PRESS ENTER TO CONTI
NUE.";A$
730 PRINT : PRINT : PRINT : PRI
NT :
740 PRINT "      "; FLASH 1;"ARE
YOU READY TO START ?"; FLASH 0
750 INPUT "YES/NO.";A$
760 IF A$="NO" THEN GO TO 90
770 GO TO 790
780 GO TO 750
790 CLS
800 PRINT TAB (8);"INITIALIZATI
ON"
810 PRINT TAB (8);"=====
=="
820 PRINT
830 PRINT "NAME.";I
840 PRINT "====="
850 PRINT
860 INPUT "NAME.....";T$(I)
870 IF T$(I,1 TO 3)="END" THEN
GO TO 1030
880 PRINT "NAME.....";T$(I)
890 INPUT "LINE(2)....";U$(I)
900 PRINT "LINE(2)....";U$(I)
910 INPUT "LINE(3)....";V$(I)
920 PRINT "LINE(3)....";V$(I)
930 INPUT "LINE(4)....";W$(I)
940 PRINT "LINE(4)....";W$(I)
950 INPUT "LINE(5)....";X$(I)
960 PRINT "LINE(5)....";X$(I)
970 INPUT "LINE(6)....";Y$(I)
980 PRINT "LINE(6)....";Y$(I)
990 INPUT "LINE(7)....";Z$(I)
1000 PRINT "LINE(7)....";Z$(I)
1010 LET I=I+1
1020 GO TO 790
1030 OPEN #4;"M";1;"ADDRESS"

```

```

1040 CLS
1050 PRINT " "; FLASH 1;"FILE BEING WRITTEN"; FLASH 0
1060 FOR J=1 TO I-1
1070 PRINT #4;T$(J)
1080 PRINT #4;U$(J)
1090 PRINT #4;V$(J)
1100 PRINT #4;W$(J)
1110 PRINT #4;X$(J)
1120 PRINT #4;Y$(J)
1130 PRINT #4;Z$(J)
1140 NEXT J
1150 PRINT #4;"EOF"
1160 CLOSE #4
1170 STOP

```

## ADDRESS.2

This is the main file management program used to handle the ADDRESS data file initialized using the program ADDRESS.1. It should be entered very carefully and then saved using the command

```
SAVE*"m";1;"ADDRESS.2"
```

### Listing

```

10 LET TEST=0
20 CLS
30 GO SUB 9800
40 DIM T$(50,30)
50 DIM U$(50,30)
60 DIM V$(50,30)
70 DIM W$(50,30)
80 DIM X$(50,30)
90 DIM Y$(50,30)
100 DIM Z$(50,30)
110 DIM F(10)
120 OPEN #4;"M";1;"ADDRESS"
130 LET I=1
140 INPUT #4;T$(I)
150 IF T$(I,1 TO 3)="EOF" THEN
GO TO 240

```

```

160 INPUT #4;U$(I)
170 INPUT #4;V$(I)
180 INPUT #4;W$(I)
190 INPUT #4;X$(I)
200 INPUT #4;Y$(I)
210 INPUT #4;Z$(I)
220 LET I=I+1
230 GO TO 140
240 CLOSE #4
250 GO SUB 1280
260 CLS
270 REM *MAIN MENU*
280 CLS
290 PRINT TAB (12);"MAIN-MENU"
300 PRINT TAB (12);"===== "
310 PRINT : PRINT : PRINT
320 PRINT "1....PRINT COMPLETE
FILE"
330 PRINT
340 PRINT "2....SEARCH AND DISP
LAY"
350 PRINT
360 PRINT "3....DELETE A RECORD
"
370 PRINT
380 PRINT "4....CHANGE A RECORD
"
390 PRINT
400 PRINT "5....ADD RECORDS"
410 PRINT
420 PRINT "6....CLOSE ADDRESS B
OOK"
430 PRINT : PRINT
440 INPUT "SELECTION=";A
450 IF A<1 OR A>6 THEN GO TO 4
40
460 GO TO A*1000
1000 CLS
1010 PRINT TAB (10);"PRINT SECTI
ON"
1020 PRINT TAB (10);"===== =====
=="
1030 PRINT : PRINT : PRINT
1040 PRINT "1....PRINT CONTENTS
TO SCREEN"

```

```

1050 PRINT
1060 PRINT "2....PRINT CONTENTS
TO PRINTER"
1070 INPUT "SELECTION=";A
1080 IF A>2 OR A<1 THEN GO TO 1
070
1090 IF A=1 THEN GO TO 1110
1100 IF A=2 THEN GO TO 1410
1110 CLS
1120 FOR J=1 TO I-1
1130 PRINT TAB (12);"FILE DUMP"
1140 PRINT TAB (12);"==== ====="
1150 PRINT "RECORD.";J
1160 PRINT
1170 PRINT T$(J)
1180 PRINT U$(J)
1190 PRINT V$(J)
1200 PRINT W$(J)
1210 PRINT X$(J)
1220 PRINT Y$(J)
1230 PRINT Z$(J)
1240 INPUT "PRESS ENTER TO CONTI
NUE.";A$
1250 CLS
1260 NEXT J
1270 GO TO 270
1280 CLS
1290 PRINT TAB (12);"FILE DUMP"
1300 PRINT TAB (12);"==== ====="
1310 PRINT : PRINT
1320 PRINT "1....ZX-PRINTER"
1330 PRINT
1340 PRINT "2....SPECIAL RS232 P
RINTER"
1350 INPUT "SELECTION=";A
1360 IF A<1 OR A>2 THEN GO TO 1
350
1370 IF A=1 THEN RETURN
1380 FORMAT "T";600
1390 OPEN #3;"T"
1400 RETURN
1410 PRINT #3;"          FILE D
UMP"
1420 PRINT #3;"          =====
====="

```

```

1430 FOR J=1 TO I-1
1440 PRINT #3;"RECORD.";J
1450 PRINT #3;"======"
1460 PRINT #3;T$(J)
1470 PRINT #3;U$(J)
1480 PRINT #3;V$(J)
1490 IF CODE (W$(J))=32 THEN GO
    TO 1570
1500 PRINT #3;W$(J)
1510 IF CODE (X$(J))=32 THEN GO
    TO 1570
1520 PRINT #3;X$(J)
1530 IF CODE (Y$(J))=32 THEN GO
    TO 1570
1540 PRINT #3;Y$(J)
1550 IF CODE (Z$(J))=32 THEN GO
    TO 1570
1560 PRINT #3;Z$(J)
1570 PRINT #3;' '
1580 NEXT J
1590 GO TO 270
2000 CLS
2010 LET RF=1
2020 PRINT TAB (7);"SEARCH AND D
ISPLAY"
2030 PRINT TAB (7);"==== = = =
====="
2040 PRINT : PRINT
2050 GO SUB 2070
2060 GO TO 2380
2070 INPUT "NAME=";N$
2080 PRINT "NAME=";N$
2090 LET RF=1
2100 PRINT
2110 PRINT " "; FLASH 1;"SEAR
CHING"; FLASH 0
2120 PRINT
2130 PRINT "FOUND"
2140 PRINT "====="
2150 PRINT
2160 FOR J=1 TO I-1
2170 FOR K=1 TO (30-LEN (N$))
2180 IF T$(J,K TO (K+LEN (N$)-1)
)=N$ THEN PRINT RF;"...";T$(J):
    LET F(RF)=J: LET RF=RF+1

```

```

2190 NEXT K
2200 NEXT J
2210 LET A=1
2220 IF RF=2 THEN GO TO 2260
2230 IF RF=1 THEN GO TO 2500
2240 INPUT "SELECTION=";A
2250 IF A<1 OR A>RF THEN GO TO
2240
2260 CLS
2270 PRINT TAB (12);"ADDRESS"
2280 PRINT TAB (12);"======"
2290 PRINT : PRINT
2300 PRINT T$(F(A))
2310 PRINT U$(F(A))
2320 PRINT V$(F(A))
2330 PRINT W$(F(A))
2340 PRINT
2350 PRINT Y$(F(A))
2360 PRINT Z$(F(A))
2370 RETURN
2380 INPUT "DO YOU REQUIRE A PRI
NT OUT.";A$
2390 IF A$="NO" THEN GO TO 270
2400 PRINT #3;TAB (12);"ADDRESS"
2410 PRINT #3;TAB (12);"======"
2420 PRINT #3;T$(F(A))
2430 PRINT #3;U$(F(A))
2440 PRINT #3;V$(F(A))
2450 PRINT #3;W$(F(A))
2460 PRINT #3;X$(F(A))
2470 PRINT #3;Y$(F(A))
2480 PRINT #3;Z$(F(A))
2490 GO TO 270
2500 PRINT FLASH 1;"NO RECORD E
XISTS"; FLASH 0
2510 INPUT "PRESS ENTER TO CONTI
NUE.";A$
2520 GO TO 270
3000 CLS
3010 LET TEST=1
3020 PRINT TAB (12);"DELETIONS"
3030 PRINT TAB (12);"======"
3040 PRINT : PRINT
3050 INPUT "NAME TO BE DELETED."
;N$

```

```

3060 PRINT "NAME.....";N$
3070 PRINT : PRINT
3080 FOR J=1 TO I-1
3090 FOR K=1 TO (30-LEN (N$))
3100 IF T$(J,K TO (K+LEN (N$))-1)
)=$ THEN GO TO 3150
3110 NEXT K : NEXT J
3120 PRINT FLASH I;"NO SUCH REC
ORD"; FLASH 0
3130 INPUT "PRESS ENTER TO CONTI
NUE.":A$
3140 GO TO 270
3150 FOR S=J TO I-1
3160 LET T$(S)=T$(S+1)
3170 LET U$(S)=U$(S+1)
3180 LET V$(S)=V$(S+1)
3190 LET W$(S)=W$(S+1)
3200 LET X$(S)=X$(S+1)
3210 LET Y$(S)=Y$(S+1)
3220 LET Z$(S)=Z$(S+1)
3230 NEXT S
3240 LET I=I-1
3250 PRINT FLASH I;"RECORD DELE
TED"; FLASH 0
3260 INPUT "PRESS ENTER TO CONTI
NUE.":A$
3270 GO TO 270
4000 CLS
4010 LET TEST=1
4020 PRINT TAB (12);"CHANGES"
4030 PRINT TAB (12);"======"
4040 PRINT : PRINT
4050 GO SUB 2070
4060 PRINT FLASH I;"CHANGES": F
LASH 0
4070 GO SUB 4090
4080 GO TO 4230
4090 PRINT "NAME.....":T$(F(A)
)
4100 INPUT "LINE(2).....":U$(F(A)
)
4110 PRINT "LINE(2).....":U$(F(A)
)
4120 INPUT "LINE(3).....":V$(F(A)
)
)

```

```

4130 PRINT "LINE(3)....";V$(F(A)
)
4140 INPUT "LINE(4)....";W$(F(A)
)
4150 PRINT "LINE(4)....";W$(F(A)
)
4160 INPUT "LINE(5)....";X$(F(A)
)
4170 PRINT "LINE(5)....";X$(F(A)
)
4180 INPUT "LINE(6)....";Y$(F(A)
)
4190 PRINT "LINE(6)....";Y$(F(A)
)
4200 INPUT "LINE(7)....";Z$(F(A)
)
4210 PRINT "LINE(7)....";Z$(F(A)
)
4220 RETURN
4230 INPUT "PRESS RETURN TO CONT
INUE.";A$
4240 GO TO 270
5000 CLS
5010 LET TEST=1
5020 PRINT TAB (9);"NEW RECORDS"
5030 PRINT TAB (9);"=== ====="
5040 PRINT : PRINT
5050 INPUT "NAME....";T$(I)
5060 PRINT "NAME.....";T$(I)
5070 LET A=1: LET F(A)=I
5080 GO SUB 4100
5090 LET I=I+1
5100 INPUT "PRESS ENTER TO CONTI
NUE.";A$
5110 GO TO 270
6000 CLS
6010 PRINT TAB (6);"CLOSE PROCED
URE"
6020 PRINT TAB (6);"=====
===="
6030 PRINT : PRINT
6040 IF TEST=0 THEN GO TO 6200
6050 PRINT FLASH 1;"WRITING NEW
FILE"; FLASH 0

```



```

6060 PRINT : PRINT
6070 ERASE "M";1;"ADDRESS"
6080 OPEN #4;"M";1;"ADDRESS"
6090 FOR J=1 TO I-1
6100 PRINT #4;T$(J)
6110 PRINT #4;U$(J)
6120 PRINT #4;V$(J)
6130 PRINT #4;W$(J)
6140 PRINT #4;X$(J)
6150 PRINT #4;Y$(J)
6160 PRINT #4;Z$(J)
6170 NEXT J
6180 PRINT #4;"EOF"
6190 CLOSE #4
6200 PRINT " "; FLASH 1;"THANK
YOU FOR USING ADDRESS"; FLASH 0
6210 PRINT " "; FLAS
H 1;"MANAGER"; FLASH 0
6220 PRINT : PRINT : PRINT
6230 PRINT " "; FLASH
1;"GOOD--BYE"; FLASH 0
6240 CLOSE #3
9000 STOP
9800 PRINT " "; FLASH 1;"WELC
OME TO ADDRESS MANAGER"; FLASH 0
9810 PRINT : PRINT : PRINT
9820 PRINT "FILE NOW LOADING"
9830 RETURN

```

## Instructions

Having reached this point, it is assumed that both programs have been entered, checked and saved on a cartridge, so that we are now ready to initialize the system and start using the address book.

### Stage 1

Load the program ADDRESS.1 (using

```
LOAD*"m";1;"ADDRESS.1")
```

and then type RUN. A series of instructions concerning how

names and addresses should be entered will appear on the screen, and these should be read carefully before continuing on to the next section.

It is observed from the introduction that all addresses are entered and saved using the same number of lines (one for the name and six for the address), with the user typing ENTER if no line is to be recorded. This overcomes the difficulties caused by addresses being of different lengths (see Fig. 7).

Mr. P. Smith, 16 Downs Lane, Swansea SW4 1TS.	Mr. J. Jones, The Den, Uplands Lane, Farnham, Surrey SU6 6TT.
--------------------------------------------------------	------------------------------------------------------------------------------

Fig. 7.

After entering a selection of records (*they do not all have to be entered at this stage*), the information can be saved on file by typing the word END in reply to the question NAME=. The information will then be transferred to a micro drive cartridge, where it will be stored in a data file called ADDRESS.

## Stage 2

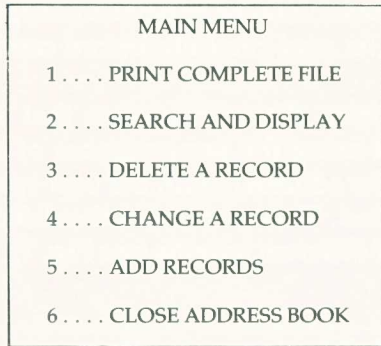
After 'ADDRESS.1' has been used to create the initial file, it will no longer be needed unless a new address book is to be set up on a second cartridge, and for this reason it should be kept in reserve.

The address book is now ready to be used. Type

```
LOAD*"m";1;"ADDRESS.2"
```

to load the main program and then enter RUN. A welcome message will appear on the screen and the computer will automatically load the current address file into memory. When this is completed, the computer will request information concerning your type of printer and then the main menu (Fig. 8) will appear on the screen.

By typing a number in the range 1-6 (anything else will be ignored), you can gain access to any of the facilities shown in Fig. 8. These features are described below.



*Fig. 8.*

### **Print Complete File**

Section 1 prints the whole file (names and addresses) onto the screen or printer. This facility is best used as a method of checking that a file is complete or for obtaining a printed list for further use.

### **Search and Display**

This section is designed to be very comprehensive, and it will work on a pattern of letters, rather than the complete name. This means that if we type the letters SM the computer will locate and display the names of all the entries containing this combination of letters: for example

- 1 . . . . SMITH.A
- 2 . . . . SMITH.B
- 3 . . . . SMALL
- 4 . . . . SMEDLEY
- 5 . . . . CHISSM

The user is then required to select which record he wishes to display by entering the appropriate number, and the computer responds by displaying the full name and address on the screen.

### **Delete a Record**

If selection 3 is made, the user is able to delete a complete record from the address book. This is achieved by typing a name or

combination of letters; the computer then searches for these as in selection 2, displaying all the possibilities on the screen. By specifying the record number, the appropriate entry will then be deleted from the file.

### **Change a Record**

This facility allows for the fact that people often change their address. Hence we will need to change the address fields of the record but leave the name field unaltered. This is achieved by locating the record as before and then inputting the new address fields from the keyboard, thus overwriting the old ones currently on the file. The method of entry is the same as in the ADDRESS.1 program, with the ENTER key being used when no line of information is required.

### **Add Records**

In the instructions for ADDRESS.1, it was stated that it was not necessary to enter all the records at the preliminary stage. This is because this section of the file management program allows new names and addresses to be added to the end of the file.

### **Close Address Book**

This is the most important section and should always be the final step when using the address book. The computer performs a check on the file to see if any alterations have been made, and if this is the case the old version is erased from the cartridge and the new, updated version saved in its place.

### **Conclusion**

The technique used in this program reads the whole file from the cartridge into the main store (RAM) where it remains while all processing is performed. This increases the speed at which the program works but limits us to the number of records that we can handle since the main store soon becomes full. An alternative technique which can be used for very large data files is discussed and demonstrated in Chapter 6.

# The Electronic Diary

The three programs contained in this chapter will provide the user with an 'electronic everlasting diary' that contains all the facilities one would expect to have with a conventional diary, as well as many extra features that would not be possible using pen and paper.

It is very difficult to list all the advantages of using a computer as a diary and the best way of appreciating them would be to type the programs and then experience the diary at first hand.

## Facilities

- 1) Full calculation of day, given the date.
- 2) Automatic increment of the date.
- 3) Full editing facilities.
- 4) Birthday/anniversary book with full editing facilities.
- 5) Birthday/anniversaries automatically displayed.
- 6) Advance warning of birthdays and anniversaries.
- 7) Diary works on a full 365- (366) day cycle with old entries automatically deleted.

The full system is very complex and requires the use of all three programs, plus a *blank* micro drive cartridge. Because of the complexity it is most important that the instructions found with each listing and at the end of the chapter are followed down to the last letter.

## DIARY.1

The program should be typed into the computer, checked very carefully and then saved on a spare cartridge using the command

```
SAVE*"m";1;"DIARY.1"
```

## Note

The program must not be saved on the actual cartridge to be used as the electronic diary at this stage.

## Listing

```

1 DIM M$(12,15)
2 DIM D(12)
3 DIM S$(7,9)
4 CLOSE #3: CLOSE #4
5 LET I$=""
10 CLS
20 PRINT TAB (6);"DIARY INITIAL
IZATION"
30 PRINT TAB (6);"=====
=====
40 PRINT : PRINT
50 PRINT "THE OBJECT OF THIS P
ROGRAM IS TO"
60 PRINT
70 PRINT "INITALIZE THE MICROD
RIVE"
80 PRINT
90 PRINT "CARTRIDGE WHICH IS T
O BE USED"
100 PRINT
110 PRINT "FOR THE 'ELECTRONIC
DIARY'."
120 PRINT
130 PRINT
140 PRINT "THIS PROGRAM MUST ON
LY BE USED"
150 PRINT
160 PRINT "ONCE AS IT WILL RE-F
ORMAT THE"
170 PRINT
180 PRINT "CARTRIDGE THUS DESTR
OYING ITS"
190 PRINT
200 PRINT "CONTENTS."
210 INPUT "PRESS ENTER TO CONTI
NUE. ";A$
220 CLS

```

```

230 PRINT TAB (6);"DIARY INITIAL
IZATION"
240 PRINT TAB (6);"=====
===== "
250 PRINT : PRINT
260 PRINT "THE INITIALIZATION PR
OCCESS WILL "
270 PRINT
280 PRINT "INVOLVE YOU ENTERING
THE DATE"
290 PRINT
300 PRINT "AFTER WHICH THE PROG
RAM WILL "
310 PRINT "TAKE OVER THE CONTRO
L OF "
320 PRINT
330 PRINT "MICRO DRIVE UNTIL TH
E PROCESS "
340 PRINT
350 PRINT "HAS BEEN COMPLETED."
360 PRINT : PRINT : PRINT
370 PRINT FLASH 1;"PLACE A NEW
CARTRIDGE IN DRIVE.1"; FLASH 0
380 INPUT "PRESS ENTER TO CONTI
NUE";A$
390 GO SUB 9000
400 CLS
410 PRINT : PRINT : PRINT : PRI
NT
420 PRINT " "; FLASH 1;"INIT
IALIZATION IN PROCESS"; FLASH 0
430 PRINT : PRINT
440 PRINT " "; FLASH
1;"PLEASE WAIT"; FLASH 0
445 FORMAT "M";1;"DIARY"
450 OPEN #4;"M";1;"DATE"
460 PRINT #4;A
470 PRINT #4;B
480 PRINT #4;C
490 PRINT #4;DAY
495 PRINT #4;DAT1
500 CLOSE #4
510 FOR I=1 TO 12
511 FOR J=1 TO D(I)

```

```

512 LET I$=I$+"*"
513 NEXT J
520 LET N$=M$(I,1 TO 3)
530 OPEN #4;"M";1;N$
560 PRINT #4;I$
580 CLOSE #4
585 LET I$=""
590 NEXT I
591 OPEN #4;"M";1;"BIRTHDAY"
592 PRINT #4;"EOF"
593 CLOSE #4
600 PRINT : PRINT : PRINT
610 PRINT " "; FLASH 1;"INITA
LIZATION IS COMPLETE"; FLASH 0
620 STOP
8000 STOP
9000 CLS
9001 LET DAY=0
9010 FOR I=1 TO 12
9020 READ M$(I),D(I)
9030 NEXT I
9035 FOR I=1 TO 7
9036 READ S$(I)
9037 NEXT I
9040 INPUT "DATE USING DD/MM/YY=
";D$
9060 LET A$=D$(1 TO 2)
9070 LET B$=D$(4 TO 5)
9075 LET C$=D$(7 TO 8)
9080 LET A=VAL (A$)
9090 LET B=VAL (B$)
9100 LET C=VAL (C$)
9110 LET A1=A-1
9120 LET B1=B-1
9130 LET C1=C-83
9135 IF C1=0 THEN GO TO 9180
9140 FOR I=1 TO C1
9150 IF (83+C1)/4=INT ((83+C1)/4
) THEN LET DAY=DAY+366: GO TO 9
170
9160 LET DAY=DAY+365
9170 NEXT I
9175 IF B1=0 THEN GO TO 9210
9180 FOR I=1 TO B1

```



```

9190 LET DAY=DAY+D(I)
9200 NEXT I
9210 LET DAY=DAY+A1
9215 LET DAT1=DAY
9220 LET CHECK=INT (DAY/7)
9230 LET DAY=(DAY-(CHECK*7))
9240 RETURN
9900 DATA "JANUARY",31,"FEBRUARY",28,
" MARCH",31,"APRIL",30,"MAY",31,
" JUNE",30
9910 DATA "JULY",31,"AUGUST",31,
"SEPTMBER",30,"OCTOBER",31,"NOVEMBER",30,
"DECEMBER",31
9920 DATA "SUNDAY","MONDAY","TUESDAY",
"WEDNESDAY","THURSDAY","FRIDAY",
" SATURDAY"

```

*Note*

Having entered, checked and saved this program on a spare cartridge, we are now ready to use this program to format and initialize our 'electronic diary'. By doing this we are then able to save the next two programs directly on the 'electronic diary' cartridge.

Load the program from the spare cartridge and then type RUN. You will then be requested to insert a blank cartridge and press ENTER. When this is done, the formatting and initialization will take place. Do not worry if the drive seems to be operating for a long time, as there are numerous files to be created and the complete process will take several minutes.

After the initialization has been completed, the program should then be saved by typing

```
SAVE*"m";1;"DIARY.1"
```

**DIARY.2**

This second program is the auto-run section of the system which will be used each day to increment the date and load the main 'electronic diary' program. It should be entered with great care, checked and then saved on the newly formatted cartridge using the command

```
SAVE*"m";1;"run" LINE 1
```

## Note

It is most important that the program name "run" is entered using small letters and not capitals.

## Listing

```

1 DIM M$(12,15)
2 DIM D(12)
3 DIM S$(7,9)
4 DIM I$(31,200)
10 CLS
20 FOR I=1 TO 12
30 READ M$(I),D(I)
40 NEXT I
50 FOR I=1 TO 7
60 READ S$(I)
70 NEXT I
80 GO SUB 8500
90 LET A=A+1
100 LET DAY=DAY+1
105 LET DAT1=DAT1+1
110 IF DAY>7 THEN LET DAY=1
120 IF B=2 AND A=29 AND C/4=INT
(C/4) THEN GO TO 150
130 IF A>D(B) THEN LET A=1: LE
T B=B+1
140 IF B>12 THEN LET B=1: LET
C=C+1
150 GO SUB 8610
155 PRINT : PRINT : PRINT
156 PRINT " "; INK 2;
FLASH 1;"PLEASE LEAVE"; FLASH 0
157 PRINT
158 PRINT " "; INK 2; FL
ASH 1;"SAVING THE DATE"; FLASH 0
; INK 0
160 ERASE "M";1;"DATE"
170 OPEN #4;"M";1;"DATE"
180 PRINT #4;A
190 PRINT #4;B
200 PRINT #4;C
210 PRINT #4;DAY
215 PRINT #4;DAT1

```

```

220 CLOSE #4
230 INPUT "PRESS ENTER TO LOAD
DIARY.";A$
235 PRINT
236 PRINT " "; INK 1; F
LASH 1;"LOADING DIARY"; FLASH 0;
INK 0
240 LOAD *"M";1;"DIARY"
250 STOP
8500 CLS
8510 PRINT " "; FLASH
1;"WELCOME TO"; FLASH 0
8520 PRINT " "; FLASH 1;"THE
ELECTRONIC DIARY"; FLASH 0
8530 PRINT : PRINT : PRINT :
8540 PRINT " "; FLASH 1
;"TODAYS DATE"; FLASH 0
8541 PLOT 50,120
8542 DRAW 150,0
8543 DRAW 0,-60
8544 DRAW -150,0
8545 DRAW 0,60
8550 OPEN #4;"M";1;"DATE"
8560 INPUT #4;A
8570 INPUT #4;B
8580 INPUT #4;C
8590 INPUT #4;DAY
8595 INPUT #4;DAT1
8600 CLOSE #4
8605 RETURN
8610 PRINT : PRINT
8614 LET T=0
8615 FOR I=1 TO LEN (S$(DAY))
8616 IF S$(DAY,I)<>" " THEN LET
T=T+1
8617 NEXT I
8618 LET T=((32-T)/2)-1
8620 PRINT TAB (T);S$(DAY)
8624 LET T=0
8625 FOR I=1 TO LEN (M$(B))
8626 IF M$(B,I)<>" " THEN LET T
=T+1
8627 NEXT I
8628 IF A>9 THEN LET T=T+1

```

```

8630 PRINT TAB (((32-(T+2))/2)-1
);A;" ";M$(B)
8635 PRINT
8640 PRINT TAB (13);1900+C
8641 PLOT 50,120
8642 DRAW 150,0
8643 DRAW 0,-60
8644 DRAW -150,0
8645 DRAW 0,60
8650 RETURN
9900 DATA "JANUARY",31,"FEBRUARY
",28,"MARCH",31,"APRIL",30,"MAY"
,31,"JUNE",30
9910 DATA "JULY",31,"AUGUST",31,
"SEPTEMBER",30,"OCTOBER",31,"NOV
EMBER",30,"DECEMBER",31
9920 DATA "SUNDAY","MONDAY","TUE
SDAY","WEDNESDAY","THURSDAY","FR
IDAY","SATURDAY"

```

*Note*

*Do not attempt to run the program at this stage.*

**DIARY.3**

This is the third, final and main program of the system, which incorporates all the major features of the 'electronic diary'. It should be entered very carefully and then saved using the command GOTO 9999. This will save the program in such a way that it can be loaded and run automatically from the second program DIARY.2.

*Listing*

```

1 DIM M$(12,15)
2 DIM D(12)
3 DIM S$(7,9)
4 LET I$=""
5 LET TEST=0: LET ERROR=0
6 LET CH=0

```

```

7 CLOSE #3: CLOSE #4
8 DIM L$(50,14)
9 DIM O$(50,11)
10 DIM P$(50,5)
11 OPEN #4;"M";1;"BIRTHDAY"
12 LET NB=1
13 INPUT #4;L$(NB)
14 IF L$(NB,1 TO 3)="EOF" THEN
GO TO 19
15 INPUT #4;O$(NB)
16 INPUT #4;P$(NB)
17 LET NB=NB+1
18 GO TO 13
19 CLOSE #4
20 FOR I=1 TO 12
30 READ M$(I),D(I)
40 NEXT I
50 FOR I=1 TO 7
60 READ S$(I)
70 NEXT I
80 OPEN #4;"M";1;"DATE"
90 INPUT #4;A
100 INPUT #4;B
110 INPUT #4;C
120 INPUT #4;DAY
125 INPUT #4;DAT1
130 CLOSE #4
140 CLS
141 LET CHANGE=0
150 PRINT TAB (11);"MAIN MENU"
160 PRINT TAB (11);"==== ====="
170 PRINT : PRINT
180 PRINT "1.....DISPLAY TODAYS
ENTRY"
190 PRINT
200 PRINT "2.....DISPLAY A FUTUR
E ENTRY"
210 PRINT
220 PRINT "3.....MAKE AN ENTRY"
230 PRINT
240 PRINT "4.....DELETE AN ENTRY
"
250 PRINT
260 PRINT "5.....ENTER BIRTHDAY
BOOK"

```

```

261 PRINT
262 PRINT "6....CHANGE THE DAT
E"
263 PRINT
264 PRINT "7....CLOSE DIARY"
270 INPUT "SELECTION=";S
280 IF S<1 OR S>7 THEN GO TO 2
70
290 GO TO (S*1000)
925 FOR I=1 TO LEN (I$)
1000 REM **TODAYS ENTRY**
1010 CLS
1020 PRINT TAB (10);"TODAYS ENTR
Y"
1030 PRINT TAB (10);"===== =====
="
1040 PRINT
1050 PRINT INVERSE 1;"DATE"; IN
VERSE 0
1060 PRINT S$(DAY)
1070 PRINT A;" ";M$(B)
1080 PRINT 1900+C
1090 PRINT
1100 PRINT INVERSE 1;"ENTRIES";
INVERSE 0
1110 LET A$=M$(B,1 TO 3)
1115 LET AS=A
1116 LET A1=A: LET B1=B: LET C1=
C
1117 LET DAY1=DAY
1120 GO SUB 9200
1130 GO TO 140
2000 REM **FUTURE ENTRY**
2010 CLS
2020 PRINT TAB (10);"FUTURE ENTR
Y"
2030 PRINT TAB (10);"===== =====
="
2040 PRINT
2050 PRINT INVERSE 1;"DATE"; IN
VERSE 0
2060 INPUT "DATE USING DD/MM/YY=
";Z$
2070 LET A1=VAL (Z$(1 TO 2))

```

```
2080 LET B1=VAL (Z$(4 TO 5))
2090 LET C1=VAL (Z$(7 TO 8))
2095 GO SUB 9000
2100 GO SUB 9600
2105 IF ERROR=1 THEN GO TO 2060
2110 PRINT S$(DAY1)
2120 PRINT A1;" ";M$(B1)
2130 PRINT 1900+C1
2140 PRINT
2150 PRINT INVERSE 1;"ENTRIES";
    INVERSE 0
2160 LET A$=M$(B1,1 TO 3)
2170 LET AS=A1
2180 GO SUB 9200
2190 GO TO 140
3010 CLS
3020 PRINT TAB (10);"NEW ENTRIES
"
3030 PRINT TAB (10);"=== =====
"
3040 PRINT
3041 PRINT INVERSE 1;"DATE"; IN
VERSE 0
3050 INPUT "DATE IN FORM DD/MM/Y
Y=";Z$
3060 LET A1=VAL (Z$(1 TO 2))
3070 LET B1=VAL (Z$(4 TO 5))
3080 LET C1=VAL (Z$(7 TO 8))
3085 GO SUB 9000
3090 GO SUB 9600
3095 IF ERROR=1 THEN GO TO 3050
3100 PRINT S$(DAY1)
3110 PRINT A1;" ";M$(B1)
3120 PRINT 1900+C1
3130 PRINT
3200 LET A$=M$(B1,1 TO 3)
3205 IF TEST=B1 THEN GO TO 3255
3210 OPEN #4;"M";1;A$
3230 INPUT #4;I$
3250 CLOSE #4
3251 LET TEST=B1
3260 LET P=1
3270 FOR I=1 TO LEN (I$)
3280 IF P=A1 THEN GO TO 3300
```

```

3290 IF I$(I)="*" THEN LET P=P+
1
3295 NEXT I
3296 STOP
3300 LET G$=I$(1 TO I-1)
3310 LET H$=I$(I TO LEN (I$))
3330 INPUT "PLEASE TYPE ENTRY=";
E$
3340 PRINT INVERSE 1;"ENTRIES";
INVERSE 0
3345 PRINT E$
3350 LET I$=G$+"/"+E$+H$
3380 INPUT "ANY MORE ENTRIES FOR
THIS DATE.";Q$
3390 IF Q$="YES" THEN GO TO 326
0
3400 INPUT "ANY MORE ENTRIES FOR
THIS MONTH.";Q$
3410 IF Q$="YES" THEN GO TO 301
0
3420 PRINT : PRINT
3430 PRINT " "; FLASH
1;"SAVING DATA"; FLASH 0
3440 ERASE "M";1;A$
3450 OPEN #4;"M";1;A$
3470 PRINT #4;I$
3490 CLOSE #4
3495 LET TEST=B1
3500 GO TO 140
4000 REM **DELETE ENTRY**
4010 CLS
4020 PRINT TAB (10);"DELETE ENTR
IES"
4030 PRINT TAB (10);"===== =====
===="
4040 PRINT
4050 INPUT "DATE IN FORM DD/MM/Y
Y=";Z$
4060 LET A1=VAL (Z$(1 TO 2))
4070 LET B1=VAL (Z$(4 TO 5))
4080 LET C1=VAL (Z$(7 TO 8))
4090 LET A$=M$(B1,1 TO 3)
4095 GO SUB 9600
4096 IF ERROR=1 THEN GO TO 4050

```



```

4100 IF TEST=B1 THEN GO TO 4140
4110 OPEN #4;"M";1;A$
4120 INPUT #4;I$
4130 CLOSE #4
4135 LET TEST=B1
4140 GO SUB 9000
4150 PRINT INVERSE 1;"DATE"; IN
VERSE 0
4160 PRINT S$(DAY1)
4170 PRINT A1;" ";M$(B1)
4180 PRINT 1900+C1
4190 PRINT
4195 LET CH=1
4196 LET AS=A1
4200 GO SUB 9245
4210 LET CH=0
4220 INPUT "ENTRY TO BE DELETED="
";EN
4240 LET P=1
4250 FOR I=1 TO LEN (I$)
4260 IF P=A1 THEN GO TO 4290
4270 IF I$(I)="*" THEN LET P=P+
1
4280 NEXT I
4290 LET G$=I$(1 TO I)
4300 LET E$=""
4310 LET J=1
4320 FOR K=I TO LEN (I$)
4330 IF I$(K)<>"/" THEN NEXT K
4340 LET J=J+1
4350 IF J<>EN+1 THEN NEXT K
4355 LET G$=I$(1 TO K-1)
4360 FOR L=K+1 TO LEN (I$)
4370 IF I$(L)<>"/" AND I$(L)<>"*
" THEN LET E$=E$+I$(L): NEXT L
4400 LET H$=I$(L TO LEN (I$))
4410 LET I$=G$+H$
4420 ERASE "M";1;A$
4430 OPEN #4;"M";1;A$
4440 PRINT #4;I$
4450 CLOSE #4
4460 GO TO 140
5001 CLS
5010 PRINT TAB (8);"BIRTHDAY BOO
K"

```

```

5015 PRINT TAB (8);"===== ==
="
5020 PRINT : PRINT
5030 PRINT "1....PRINT COMPLETE
BOOK"
5040 PRINT
5050 PRINT "2....INPUT A NEW DAT
E"
5055 PRINT
5060 PRINT "3....DELETE AN ENTRY
"
5070 PRINT
5080 PRINT "4....SEARCH AND PRIN
T A DATE"
5090 PRINT
5091 PRINT "5....RETURN TO MAIN
MENU"
5092 INPUT "SELECTION=";S
5093 IF S<1 OR S>5 THEN GO TO 5
092
5095 GO TO 5000+(S*100)
5100 CLS
5105 PRINT TAB (10);"COMPLETE LI
ST"
5110 PRINT TAB (10);"===== ==
=="
5115 PRINT "NAME";TAB (14);"EVEN
T";TAB (26);"DATE"
5120 PRINT "====";TAB (14);"====
=";TAB (26);"===="
5125 FOR K=1 TO NB-1
5130 PRINT L$(K);TAB (14);O$(K);
TAB (26);P$(K)
5135 NEXT K
5140 INPUT "PRESS ENTER TO CONTI
NUE.";Z$
5150 GO TO 5000
5200 CLS
5210 PRINT TAB (10);"NEW ENTRY"
5220 PRINT TAB (10);"=== ====="
5230 PRINT : PRINT
5240 INPUT "NAME=";L$(NB)
5245 PRINT "NAME=";L$(NB)
5246 PRINT

```

```

5250 INPUT "OCCASION=";O$(NB)
5255 PRINT "OCCASION=";O$(NB)
5256 PRINT
5260 INPUT "DATE USING DD/MM=";P
$(NB)
5265 PRINT "DATE=";P$(NB)
5266 LET NB=NB+1
5270 INPUT "PRESS ENTER TO CONTI
NUE.";Z$
5280 GO TO 5000
5300 CLS
5305 PRINT TAB (10);"DELETE ENTR
Y"
5310 PRINT TAB (10);"===== =
="
5320 PRINT : PRINT
5325 INPUT "NAME=";N$
5330 FOR K=1 TO NB-1
5335 IF L$(K,1 TO LEN (N$))=N$ T
HEN GO TO 5350
5340 NEXT K
5350 PRINT "NAME=";L$(K)
5355 PRINT
5360 PRINT "OCCASION=";O$(K)
5365 PRINT
5370 PRINT "DATE=";P$(K)
5380 INPUT "IS THIS THE CORRECT
ENTRY.";Z$
5390 IF Z$="YES" THEN GO SUB 58
00
5391 IF DEL=1 THEN LET DEL=0: G
O TO 5395
5392 INPUT "DO YOU WISH TO TRY A
GAIN.";Z$
5394 IF Z$="YES" THEN GO TO 530
0
5395 INPUT "PRESS ENTER TO CONTI
NUE.";Z$
5396 GO TO 5000
5400 CLS
5410 PRINT TAB (10);"SEARCH & DI
SPLAY"
5420 PRINT TAB (10);"===== =
=====
```

```

5430 PRINT : PRINT
5440 INPUT "NAME=";N$
5450 FOR J=1 TO NB-1
5455 IF L$(J,1 TO LEN (N$))=N$ T
HEN GO TO 5480
5460 NEXT J
5465 PRINT " "; FLASH 1;"N
O SUCH ENTRY"; FLASH 0
5470 GO TO 5495
5480 PRINT "NAME=";L$(J)
5482 PRINT
5484 PRINT "OCCASION=";O$(J)
5486 PRINT
5488 PRINT "DATE=";P$(J)
5495 INPUT "PRESS ENTER TO CONTI
NUE.";Z$
5496 GO TO 5000
5500 CLS
5510 PRINT TAB (10);"CLOSE BOOK"
5520 PRINT TAB (10);"===== "
5530 PRINT : PRINT
5540 PRINT " "; FLASH 1;"S
AVING TO TAPE"; FLASH 0
5545 ERASE "M";1;"BIRTHDAY"
5550 OPEN #4;"M";1;"BIRTHDAY"
5560 FOR J=1 TO NB-1
5570 PRINT #4;L$(J)
5580 PRINT #4;O$(J)
5590 PRINT #4;P$(J)
5600 NEXT J
5610 PRINT #4;"EOF"
5620 CLOSE #4
5630 GO TO 140
5800 FOR J=K TO NB-1
5810 LET L$(J)=L$(J+1)
5820 LET O$(J)=O$(J+1)
5830 LET P$(J)=P$(J+1)
5840 NEXT J
5850 LET NB=NB-1
5860 PRINT : PRINT
5870 PRINT " "; FLASH
1;"ENTRY DELETED"; FLASH 0
5875 LET DEL=1
5880 RETURN

```

```

5999 STOP
6000 REM *CHANGE DATE*
6010 CLS
6020 PRINT TAB (10);"CHANGE DATE
"
6030 PRINT TAB (10);"===== =====
"
6040 PRINT
6050 PRINT INVERSE 1;"DATE"; IN
VERSE 0
6060 INPUT "DATE USING DD/MM/YY=
";Z$
6070 LET A1=VAL (Z$(1 TO 2))
6080 LET B1=VAL (Z$(4 TO 5))
6090 LET C1=VAL (Z$(7 TO 8))
6100 GO SUB 9000
6110 PRINT S$(DAY1)
6120 PRINT A1;" ";M$(B1)
6130 PRINT 1900+C1
6140 PRINT : PRINT
6150 PRINT INVERSE 1;"SAVING NE
W DATE"; INVERSE 0
6160 ERASE "M";1;"DATE"
6170 OPEN #4;"M";1;"DATE"
6180 PRINT #4,A1
6190 PRINT #4;B1
6200 PRINT #4;C1
6210 PRINT #4;DAY1
6215 PRINT #4;DAT2
6220 CLOSE #4
6230 IF B<>B1 THEN GO TO 6470
6240 FOR U=A TO A1-1
6250 LET P=1
6270 FOR I=1 TO LEN (I$)
6280 IF P=U THEN GO TO 6310
6290 IF I$(I)="*" THEN LET P=P+
1
6300 NEXT I
6310 LET G$=I$(1 TO I-1)
6320 LET E$=""
6330 FOR J=I TO LEN (I$)
6340 IF I$(J)="*" THEN GO TO 63
60
6350 NEXT J

```

```

6360 LET H$=I$(J TO LEN (I$))
6370 LET I$=G$+H$
6380 NEXT U
6390 LET F$=M$(B1,1 TO 3)
6400 ERASE "M";1;F$
6410 OPEN #4;"M";1;F$
6420 PRINT #4;I$
6430 CLOSE #4
6440 LET A=A1: LET B=B1: LET C=C
1: LET DAY=DAY1
6450 GO SUB 9000
6460 LET DAY=DAY1: LET DAT1=DAT2
6465 GO TO 6910
6470 IF ABS (B-B1)>1 THEN GO TO
6900
6471 LET P=1
6472 FOR I=1 TO LEN (I$)
6473 IF P=A THEN GO TO 6500
6480 IF I$(I)="*" THEN LET P=P+
1
6490 NEXT I
6500 LET G$=(I$(1 TO I-1))
6510 LET E$=""
6520 FOR J=I TO D(B)
6530 LET E$=E$+"*"
6540 NEXT J
6550 LET I$=G$+E$
6560 LET F$=M$(B,1 TO 3)
6570 ERASE "M";1;F$
6580 OPEN #4;"M";1;F$
6590 PRINT #4;I$
6600 CLOSE #4
6610 LET F$=M$(B1,1 TO 3)
6620 OPEN #4;"M";1;F$
6630 INPUT #4;I$
6640 CLOSE #4
6645 LET E$=""
6650 FOR I=1 TO (A1-1)
6660 LET E$=E$+"*"
6670 NEXT I
6680 LET P=1
6690 FOR I=1 TO LEN (I$)
6700 IF P=A1 THEN GO TO 6730
6710 IF I$(I)="*" THEN LET P=P+
1

```

```
6720 NEXT I
6730 LET G$=I$(I TO LEN (I$))
6740 LET I$=E$+G$
6745 ERASE "M";1;F$
6750 OPEN #4;"M";1;F$
6760 PRINT #4;I$
6770 CLOSE #4
6780 LET A=A1: LET B=B1: LET C=C
1: GO SUB 9000
6790 LET DAY=DAY1: LET DAT1=DAT2
6800 GO TO 140
6900 PRINT INVERSE 1;"CHANGE NO
T POSSIBLE"; INVERSE 0
6910 INPUT "PRESS ENTER TO CONTI
NUE.";Z$
6920 GO TO 140
9000 LET DAY1=0
9010 LET A2=A1-1
9020 LET B2=B1-1
9030 LET C2=C1-83
9040 IF C2=0 THEN GO TO 9100
9050 FOR I=1 TO C2
9060 IF (83+C2)/4=INT ((83+C2)/4
) THEN LET DAY1=DAY1+366: GO TO
9080
9070 LET DAY1=DAY1+365
9080 NEXT I
9090 IF B2=0 THEN GO TO 9130
9100 FOR I=1 TO B2
9110 LET DAY1=DAY1+D(I)
9120 NEXT I
9130 LET DAY1=DAY1+A2
9135 LET DAT2=DAY1
9140 LET CHECK=INT (DAY1/7)
9150 LET DAY1=(DAY1-(CHECK*7))
9151 IF DAY1=0 THEN LET DAY1=7
9160 RETURN
9200 IF TEST=B1 THEN GO TO 9245
9205 OPEN #4;"M";1;A$
9220 INPUT #4;I$
9240 CLOSE #4
9241 LET TEST=B1
9245 LET P=1
9246 LET E=1
```

```

9247 IF AS=1 THEN LET K=2: GO TO 9270
9250 FOR I=1 TO LEN (I$)
9251 IF P=AS THEN GO TO 9260
9255 IF I$(I)="*" THEN LET P=P+1
9257 NEXT I
9260 LET K=I+1
9270 LET E$=""
9275 IF I$(K-1)="*" THEN GO TO 9380
9280 LET E=1
9290 FOR J=K TO LEN (I$)
9300 IF I$(J)="*" THEN GO TO 9392
9310 IF I$(J)<>"/" THEN LET E$=E$+I$(J)
9320 IF I$(J)="/" AND J<>1 THEN GO TO 9340
9330 NEXT J
9340 PRINT E;"...";E$
9350 LET K=J+1
9360 LET E=E+1
9365 LET E$=""
9370 GO TO 9290
9380 PRINT : PRINT
9381 GO SUB 9500
9385 IF E>1 THEN GO TO 9396
9390 PRINT FLASH 1;"NO ENTRIES RECORDED"; FLASH 0: GO TO 9400
9392 PRINT E;"...";E$
9393 LET E=E+1
9395 GO SUB 9500
9396 IF CH=1 THEN RETURN
9400 INPUT "PRESS ENTER TO CONTINUE. ";B$
9410 RETURN
9500 REM **BIRTHDAY CHECK**
9510 FOR J=1 TO NB-1
9520 IF VAL (P$(J,4 TO 5))=B1 AND VAL (P$(J,1 TO 2))=A1 THEN PRINT E;"...";: GO SUB 9700
9525 IF VAL (P$(J,4 TO 5))<>B1 THEN GO TO 9530

```

I



```

9526 IF DAY1=6 AND VAL (P$(J,1 T
O 2))=A1 THEN PRINT E;"...POST
";: GO SUB 9700
9527 IF DAY1>=2 AND DAY1<=5 AND
VAL P$(J,1 TO 2)=A1+1 THEN PRIN
T E;"...POST ";: GO SUB 9700
9528 IF DAY1=7 AND VAL (P$(J,1 T
O 2))=A1+2 THEN PRINT E;"...POS
T ";: GO SUB 9700
9530 IF VAL (P$(J,4 TO 5))<>B1+1
THEN GO TO 9539
9531 IF DAY1>=2 AND DAY1<=5 AND
A1=D(B1) AND VAL (P$(J,1 TO 2))=
1 THEN PRINT E;"...POST ";: GO
SUB 9700
9532 IF DAY1=7 AND A1=D(B1) AND
VAL (P$(J,1 TO 2))=2 THEN PRINT
E;"...POST ";: GO SUB 9700
9533 IF DAY1=7 AND A1=D(B1-1) AN
D VAL (P$(J,1 TO 2))=1 THEN PRI
NT E;"...";"POST ";: GO SUB 9700
9539 NEXT J
9540 RETURN
9600 REM *DATE CHECK*
9601 IF DAT2<DAT1 THEN LET ERRO
R=1: RETURN
9602 IF DAT2-DAT1>365 THEN LET
ERROR=1: RETURN
9605 IF B1=2 THEN GO TO 9640
9610 IF A1>D(B1) THEN LET ERROR
=1: RETURN
9620 LET ERROR=0
9630 RETURN
9640 IF C1/4<>INT (C1/4) AND A1=
29 THEN LET ERROR=1: RETURN
9649 IF A1=29 THEN LET ERROR=0:
RETURN
9650 IF A1>28 THEN LET ERROR=1:
RETURN
9660 LET ERROR=0
9670 RETURN
9700 FOR G=1 TO LEN (L$(J))
9710 IF L$(J,G)=" " AND L$(J,G+1
)=" " THEN GO TO 9740

```

```

9720 PRINT L$(J,G);
9730 NEXT G
9740 PRINT "'S ";O$(J)
9745 LET E=E+1
9750 RETURN
9800 DATA "JANUARY",31,"FEBRUARY
",28,"MARCH",31,"APRIL",30,"MAY"
,31,"JUNE",30
9810 DATA "JULY",31,"AUGUST",31,
"SEPTEMBER",30,"OCTOBER",31,"NOV
EMBER",30,"DECEMBER",31
9820 DATA "SUNDAY","MONDAY","TUE
SDAY","WEDNESDAY","THURSDAY","FR
IDAY","SATURDAY"
9999 SAVE *"M";1;"DIARY" LINE 1

```

## Instructions

As mentioned at the beginning of this chapter, one of the facilities included in this system is that the date is automatically increased from one day to the next. However, this will only occur if the diary is used on a regular daily basis.

Each day, when the diary is used for the first time, the user should place the cartridge in the drive and type NEW followed by RUN. The second program DIARY.2 will then be executed, the date will be incremented and a display such as that shown in Fig. 9 will appear on the screen.



Fig. 9.

After a period of about 2 minutes, during which time the various files are updated, the main section DIARY.3 will automatically be loaded into the computer and the main menu will appear on the screen.

It is most important to remember that by using the NEW/RUN sequence the computer assumes that another day has passed, so these commands must be employed *once per day*. If the diary is needed for a second or third time during the day, it should be entered in the normal manner using the command

```
LOAD*“m”;1;“DIARY”
```

When the diary is opened, either directly using LOAD\* or indirectly using the NEW/RUN sequence, the main menu shown in Fig. 10 will appear on the screen and the user will be requested to make a selection in the range 1-7.



Fig. 10.

### Display Today's Entry

If the user selects option 1, then the computer will display all today's entries on the screen using the format shown in Fig. 11. On completion of this section pressing ENTER will allow access to the main menu again.



Fig. 11.

## Display a Future Entry

This section enables us to look at any of the 365 days during the next year (366 if it is a leap year) and view at a glance our commitments for that day. To gain access to this information the user must first input the date in the form 04/07/83. The computer will then calculate the day of the week, load the information from file and display it on the screen (see Fig. 12). The length of time taken by the computer to locate and display the data will depend on the date, but should never exceed 15 seconds.



Fig. 12.

## Make an Entry

In this section, the user is able to make an entry in the diary for any day during the next year. Firstly, the date must be entered using the format DD/MM/YY. This is then verified by the computer which calculates the day of the week, displaying the information on the screen. The user then types his entries for that day and, on completion, the new information is saved on the cartridge with the computer display returning to the main menu.

## Delete an Entry

This works in much the same way as option 3, with the obvious exception that entries are removed instead of being inserted. After the date has been entered, all the entries for the day in question will be displayed on the screen and a deletion can be made by typing the number of the entry to be removed.

**Enter Birthday Book**

As shown in Fig. 13, the birthday book is a program within a program and contains all the facilities one would expect to find in any file-handling routine.

BIRTHDAY BOOK	
1 . . . .	PRINT COMPLETE BOOK
2 . . . .	INPUT A NEW DATE
3 . . . .	DELETE AN ENTRY
4 . . . .	SEARCH AND PRINT A DATE
5 . . . .	RETURN TO MAIN MENU

*Fig. 13.*

Although termed a 'birthday book', it is designed for both birthdays and anniversaries or any event that takes place on the same day each year. The individual options within this program are obvious and full instructions are displayed during execution. Attention is drawn, however, to option 1, using which the complete book can be listed and the user can check the information.

COMPLETE LIST		
NAME	EVENT	DATE
MELISSA	BIRTHDAY	05/07
JOSIE	BIRTHDAY	13/04
RICHARD	BIRTHDAY	09/01
GINA	BIRTHDAY	08/09
FRED	ANNIVERSARY	02/03

*Fig. 14.*

The birthday file itself is an integral part of the diary since information concerning the various occasions will be listed automatically when a day's entries are displayed. Advance warning will also be given with regards to the posting of cards, with problems such as no delivery or collection on Sundays taken into account.

## **Change the Date**

If the diary is not used on a day-to-day basis then the date will not be increased and the auto-run sequence cannot be employed. Option 6 allows for such occasions, with the user being able to change the date and the computer adjusting all the files accordingly.

### *Note*

*This facility will not work if the diary is left unused for a period exceeding two months.*

## **Close Diary**

The final option is selected when the user has finished with the diary. The computer checks that all files have been correctly updated, rewriting them where necessary, and the system is shut down.

## **Conclusion**

If used in the correct manner, as documented above and described within the programs, this package will provide the user with an everlasting diary containing full information for the next 365 (366) days. But remember the computer jargon garbage in/garbage out, indicating that a computer is only as efficient as the data it is given.

# Large Data Files

Although we may be using a 48K Spectrum, we do not have 48 kilobytes available for the various data files being used, since a large portion of this total memory capacity will be required for the program and for 'workspace' (i.e. an area of memory used for processing the information obtained from the files). This may leave only a few kilobytes for file storage. If this is the case and we are employing *very large files*, then the techniques used in the last three chapters will not work as there is simply not enough room to load the complete file into main store for processing and interrogation. Under these circumstances, it would be necessary to extract only the required records from the file, passing over all those which are deemed to be superfluous at that time.

The advantage of this system is that we would be able to access enormous data bases (files) which far exceed the total memory capacity of the machine. However, what we gain in capacity we lose in speed, since the process of accessing a record from a cartridge takes very much longer than accessing data from main store.

The techniques employed in the handling of large data bases are well understood and are often used on larger, more sophisticated computers using one or preferably two disc drive units. For us, the limitations of the micro drive operating system and the fact that a file cannot be overwritten (it must be erased first) cause many problems. These can be overcome only by using some sophisticated routines which are demonstrated in the following programs.

## *Example 4*

The following program is used to set up an initial data base, in computer terminology, which can be manipulated by a second program (see Example 5). Although the original file is small, the programs are designed so that the file could be extended as

desired, the only limiting factor being the physical capacity of the cartridge being used.

```

10 CLS
20 DIM A$(10,10)
30 DIM B$(10,80)
40 FOR I=1 TO 10
50 READ A$(I),B$(I)
60 NEXT I
70 LET N$="1"
80 OPEN #4;"M";1;"NUMBER"
90 PRINT #4;N$
100 CLOSE #4
110 LET Y$="CBASE."+N$
120 OPEN #4;"M";1;Y$
130 FOR I=1 TO 10
140 PRINT #4;A$(I)
150 PRINT #4;B$(I)
160 NEXT I
170 PRINT #4;"EOF"
180 CLOSE #4
190 STOP
200 DATA "HEX","A NUMBER SYSTEM
USING BASE 16"
210 DATA "BIT","A SINGLE BINARY
CHARACTER"
220 DATA "BYTE","AN 8-BIT NUMBE
R"
230 DATA "CPU","CENTRAL PROCESS
ING UNIT"
240 DATA "RAM","RANDOM ACCESS M
EMORY"
250 DATA "ROM","READ ONLY MEMOR
Y"
260 DATA "PROM","PROGRAMMABLE R
EAD ONLY MEMORY"
270 DATA "GATE","A TINY ELECTRI
CAL CIRCUIT"
280 DATA "ALGORITHM","A SERIES
OF STEPS WHICH IF TAKEN TOGETHER
REPRESENT THE SOLUTION TO A PRO
BLEM"
290 DATA "FLOW CHART","A DIAGRA
M USED TO REPRESENT AN ALGORITHM
"

```



## COMMENTS

Lines 10-30	Initialize the variables.
Lines 40-60	Obtain information for data base from data statements.
Line 70	Assign present number to be used for C.BASE file.
Lines 80-100	Write a small file (containing one field) to record the current number of the C.BASE file.
Lines 110-180	Write C.BASE file onto cartridge.

*Example 5*

This second program is used to interrogate the computer data base set up in Example 4. The user is asked to input the terms about which he requires further information. The data base is then opened and searched in order to find the relevant information, with only the required records being entered into the computer's main store, thus economizing on the use of memory.

When the whole data base has been searched, the information is displayed together with a list of terms not included in the file. The user is then requested to locate the missing information from some other source and enter it into the computer so that the data base may be updated.

It is obvious that as the program becomes more widely used the data base will become increasingly large, eventually exceeding the 48K of the machine. However, this is of no significance since even during the updating stage the whole file is never loaded into main memory.

```

10 CLS
20 LET TEST=0
30 DIM A$(10,10)
40 DIM B$(10,80)
50 DIM C$(10,10)
60 DIM D$(10,80)
70 DIM E$(10,10)
80 DIM F$(10,80)
90 PRINT TAB (8);"COMPUTER DAT
A BASE"
100 PRINT TAB (8);"===== ===
= ====="

```

```

110 PRINT
120 FOR I=1 TO 10
130 INPUT "TERM=";C$(I)
140 IF C$(I,1 TO 3)="END" THEN
GO TO 180
150 PRINT "TERM.";I;"=";C$(I)
160 LET I=I+1
170 GO TO 130
180 CLS
190 PRINT TAB (8);"COMPUTER DAT
A BASE"
200 PRINT TAB (8);"=====
= ====="
210 PRINT
220 PRINT INVERSE 1;"LOADING F
ILE"; INVERSE 0
230 PRINT
240 OPEN #4;"M";1;"NUMBER"
250 INPUT #4;N$
260 CLOSE #4
270 LET Y$="CBASE."+N$
280 OPEN #4;"M";1;Y$
290 LET J=1
300 INPUT #4;A$(J)
310 IF A$(J,1 TO 3)="EOF" THEN
GO TO 390
320 INPUT #4;B$(J)
330 FOR K=1 TO I-1
340 IF C$(K)=A$(J) THEN LET D$
(K)=B$(J): GO TO 370
350 NEXT K
360 GO TO 300
370 LET J=J+1
380 GO TO 300
390 CLOSE #4
400 CLS
410 IF J=I THEN LET TEST=1
420 LET C=1
430 FOR J=1 TO I-1
440 CLS
450 IF D$(J,1 TO 2)=" " THEN
GO TO 550
460 PRINT TAB (8);"COMPUTER DAT
A BASE"

```

```

470 PRINT TAB (8);"===== ===
= ====="
480 PRINT
490 PRINT "TERM=";C$(J)
500 PRINT
510 PRINT "DEFINITION"
520 PRINT "===== "
530 PRINT D$(J)
540 INPUT "PRESS ENTER TO CONTI
NUE.";Z$
550 NEXT J
560 IF TEST=1 THEN STOP
570 CLS
580 PRINT TAB (8);"COMPUTER DAT
A BASE"
590 PRINT TAB (8);"===== ===
= ====="
600 PRINT
610 PRINT "INFORMATION NOT AVAI
LABLE ON THE"
620 PRINT "FOLLOWING. PLEASE FI
ND DATA AND"
630 PRINT "INPUT DEFINITION WHE
RE POSSIBLE."
640 PRINT "PRESS "; INK 2;"ENTE
R"; INK 0;"IF NO DATA AVAILABLE"
650 PRINT
660 LET S=1
670 FOR J=1 TO I-1
680 IF D$(J,1)=" " THEN GO SUB
1000
690 NEXT J
700 LET M$=N$
710 IF N$="1" THEN LET N$="2":
GO TO 730
720 LET N$="1"
730 ERASE "M";1;"NUMBER"
740 OPEN #4;"M";1;"NUMBER"
750 PRINT #4;N$
760 CLOSE #4
770 LET W$="CBASE."+N$
780 OPEN #4;"M";1;Y$
790 OPEN #5;"M";1;W$

```

```

800 INPUT #4;H$
810 IF H$(1 TO 3)="EOF" THEN G
O TO 860
820 INPUT #4;I$
830 PRINT #5;H$
840 PRINT #5;I$
850 GO TO 800
860 FOR I=1 TO S-1
870 PRINT #5;E$(I)
880 PRINT #5;F$(I)
890 NEXT I
900 PRINT #5;"EOF"
910 CLOSE #4: CLOSE #5
920 PRINT : PRINT
930 ERASE "M";1;Y$
940 STOP
1000 PRINT "TERM. ";C$(J)
1010 INPUT "DEFINITION=";F$(S)
1020 PRINT "DEFINITION=";F$(S)
1030 LET E$(S)=C$(J)
1040 LET S=S+1
1050 RETURN

```

## COMMENTS

- Lines 10-100 Initialize the variables and print the title.
- Lines 110-160 The user inputs up to a maximum of ten terms of which he requires the definitions.
- Lines 170-360 Computer inputs the data base code number from the file NUMBER and then opens the data base, extracting the required information only.
- Lines 370-480 All known terms are displayed on the screen with the relevant definitions.
- Lines 490-710 Computer searches to find what information is absent from the data base.
- Lines 720-840 The old file is rewritten under a new name and the new records are added on to the end.
- Lines 1000-1050 A subroutine in which the user is requested to find and enter new definitions required by the data base.

## Conclusion

The technique used here, and in the 'translator' program in chapter 7, is only of use when memory space is the major consideration. If the time element is important, it is better to transfer all, or as much as possible, of a file into memory.

# English/French Translator

The two programs in this chapter are used to create and then manage an English/French data base. It is initialized in the program TRANSLATOR.1, in which a series of English words and their French equivalents are read from data and then saved on a cartridge under the name T.BASE. Then, by using the second program, the user is able to enter a series of words in either English or French and the computer will extract the equivalent word in the other language from the data base. If the word is not found, then the user is requested to find the information from some other source so that the data base can be extended for future use.

It is very likely that after a short time the file would become very large and therefore the techniques discussed in Chapter 6 are employed.

## TRANS.1

This initialization program should be typed into the computer, checked and then saved using the command

```
SAVE*"m";1;"TRANS.1"
```

### *Listing*

```
10 CLS
20 DIM A$(40,20)
30 DIM B$(40,20)
40 CLS
50 PRINT TAB (10);"INITIALIZAT
ION"
```

```

60 PRINT TAB (10);"=====
=== "
70 PRINT : PRINT
80 FOR I=1 TO 20
90 READ A$(I),B$(I)
100 NEXT I
110 LET N$="1"
120 LET Y$="T.BASE"+N$
130 OPEN #4;"M";1;"NUMBER.1"
140 PRINT #4;N$
150 CLOSE #4
160 OPEN #4;"M";1;Y$
170 FOR I=1 TO 20
180 PRINT #4;A$(I)
190 PRINT #4;B$(I)
200 NEXT I
210 PRINT #4;"EOF"
220 CLOSE #4
230 STOP
240 DATA "LOVE","AMOUR"
250 DATA "MAN","HOMME"
260 DATA "MEMORY","MEMOIRE"
270 DATA "LAD","GARCON"
280 DATA "LADY","DAME"
290 DATA "LAMP","LAMPE"
300 DATA "FIVE","CINQ"
310 DATA "FIX","EMBARRAS"
320 DATA "FLAG","GLAIEUL"
330 DATA "EGG","OEUF"
340 DATA "DUCK","CANARD"
350 DATA "DUMP","COUP"
360 DATA "OUR","NOTRE"
370 DATA "OUTSIDE","DEHORS"
380 DATA "PAPER","PAPIER"
390 DATA "PEACH","PECHE"
400 DATA "PAVE","PAVER"
410 DATA "PEN","ENCLOS"
420 DATA "PENCIL","CRAYON"
430 DATA "ROPE","CORDE"

```

## TRANS.2

This program is the interrogation and updating part of the system and is used to handle T.BASE data base created in the first

program. It should be entered very carefully, checked and then saved using the command

```
SAVE*"m";1;"TRANS.2"
```

*Listing*

```

10 CLS
20 LET CH=0
30 LET Q=1
40 DIM A$(20,20)
50 DIM B$(20,20)
60 DIM C$(20,20)
70 DIM D$(20,20)
80 DIM E$(20,20)
90 DIM F$(20,20)
100 OPEN #4;"M";1;"NUMBER.1"
110 INPUT #4;N$
120 CLOSE #4
130 CLS
140 PRINT TAB (10);"TRANSLATOR"
150 PRINT TAB (10);"=====
160 PRINT : PRINT
170 PRINT "1....ENGLISH TO FREN
CH"
180 PRINT
190 PRINT "2....FRENCH TO ENGLI
SH"
200 INPUT "SELECTION=";A
210 IF A<1 OR A>2 THEN GO TO 2
00
220 GO TO (A*1000)
1000 REM *ENGLISH/FRENCH*
1010 CLS
1020 PRINT TAB (8);"ENGLISH TO F
RENCH"
1030 PRINT TAB (8);"===== == =
=====
1040 PRINT
1050 LET I=1
1060 PRINT "INPUT WORDS IN ENGLI
SH. "
1070 PRINT "PRESS 99 TO FINISH."
1080 PRINT
1090 PRINT INVERSE 1;"WORDS"; I
NVERSE 0

```



```

1100 INPUT "WORD=";C$(I)
1110 IF C$(I,1 TO 2)="99" THEN
GO TO 1150
1120 PRINT I;"...";C$(I)
1130 LET I=I+1
1140 GO TO 1100
1150 LET Y$="T.BASE"+N$
1160 OPEN #4;"M";1;Y$
1170 INPUT #4;S$
1180 IF S$(1 TO 3)="EOF" THEN G
O TO 1240
1190 INPUT #4;T$
1200 FOR J=1 TO I-1
1210 IF C$(J)=S$ AND D$(J,1)=" "
THEN LET D$(J)=T$: LET Q=Q+1:
GO TO 1170
1220 NEXT J
1230 GO TO 1170
1240 CLOSE #4
1250 IF Q=I THEN LET CH=1
1260 CLS
1270 PRINT TAB (8);"ENGLISH TO F
RENCH"
1280 PRINT TAB (8);"===== == =
===== "
1290 PRINT
1300 FOR J=1 TO I-1
1310 IF D$(J,1 TO 2)=" " THEN
GO TO 1350
1320 PRINT "WORD=";C$(J)
1330 PRINT "TRANSLATION=";D$(J)
1340 PRINT
1350 NEXT J
1360 INPUT "PRESS ENTER TO CONTI
NUE. ";Z$
1370 IF CH=1 THEN GO TO 130
1380 CLS
1390 PRINT TAB (8);"ENGLISH TO F
RENCH"
1400 PRINT TAB (8);"===== == =
===== "
1410 PRINT
1420 PRINT "INFORMATION NOT AVAI
LABLE ON THE"

```

```

1430 PRINT "FOLLOWING. PLEASE RE
SEARCH AND"
1440 PRINT "INPUT TRANSLATIONS W
HERE EVER"
1450 PRINT "POSSIBLE. PRESS ENTE
R IF NO"
1460 PRINT "TRANSLATION IS AVAIL
ABLE."
1470 PRINT
1480 LET S=1
1490 FOR J=1 TO I-1
1500 IF D$(J,1)<>" " THEN GO TO
1570
1510 PRINT "WORD=";C$(J)
1520 INPUT "TRANSLATION=";F$(S)
1530 PRINT "TRANSLATION=";F$(S)
1540 PRINT
1550 LET E$(S)=C$(J)
1560 LET S=S+1
1570 NEXT J
1580 IF CH=0 THEN GO TO 3000
1590 STOP
2000 REM *FRENCH/ENGLISH*
2010 CLS
2020 PRINT TAB (8);"FRENCH TO EN
GLISH"
2030 PRINT TAB (8);"===== == ==
====="
2040 PRINT
2050 LET I=1
2060 PRINT "INPUT WORDS IN FRENC
H. "
2070 PRINT "PRESS 99 TO FINISH."
2080 PRINT
2090 PRINT INVERSE 1;"WORDS"; I
NVERSE 0
2100 INPUT "WORD=";D$(I)
2110 IF D$(I,1 TO 2)="99" THEN
GO TO 2150
2120 PRINT I;"...";D$(I)
2130 LET I=I+1
2140 GO TO 2100
2150 LET Y$="T.BASE"+N$
2160 OPEN #4;"M";1;Y$
2170 INPUT #4;S$

```

```

2180 IF S$(1 TO 3)="EOF" THEN G
O TO 2240
2190 INPUT #4;T$
2200 FOR J=1 TO I-1
2210 IF D$(J)=T$ AND C$(J,1)=" "
THEN LET C$(J)=S$: LET Q=Q+1:
GO TO 2170
2220 NEXT J
2230 GO TO 2170
2240 CLOSE #4
2250 IF Q=I THEN LET CH=1
2260 CLS
2270 PRINT TAB (8);"FRENCH TO EN
GLISH"
2280 PRINT TAB (8);"===== == ==
===== "
2290 PRINT
2300 FOR J=1 TO I-1
2310 IF C$(J,1 TO 2)=" " THEN
GO TO 2350
2320 PRINT "WORD=";D$(J)
2330 PRINT "TRANSLATION=";C$(J)
2340 PRINT
2350 NEXT J
2360 INPUT "PRESS ENTER TO CONTI
NUE.";Z$
2370 IF CH=1 THEN GO TO 130
2380 CLS
2390 PRINT TAB (8);"FRENCH TO EN
GLISH"
2400 PRINT TAB (8);"===== == ==
===== "
2410 PRINT
2420 PRINT "INFORMATION NOT AVAI
LABLE ON THE"
2430 PRINT "FOLLOWING. PLEASE RE
SEARCH AND"
2440 PRINT "INPUT TRANSLATIONS W
HEREEVER"
2450 PRINT "POSSIBLE. PRESS ENTE
R IF NO"
2460 PRINT "TRANSLATION IS AVAIL
ABLE."
2470 PRINT

```

```
2480 LET S=1
2490 FOR J=1 TO I-1
2500 IF C$(J,1)<>" " THEN GO TO
  2570
2510 PRINT "WORD=";D$(J)
2520 INPUT "TRANSLATION=";E$(S)
2530 PRINT "TRANSLATION=";E$(S)
2540 PRINT
2550 LET F$(S)=D$(J)
2560 LET S=S+1
2570 NEXT J
2580 IF CH=0 THEN GO TO 3000
2590 STOP
3000 REM *UPDATE DATA BASE*
3010 PRINT
3020 PRINT INVERSE 1;"SAVING FI
LE"; INVERSE 0
3030 LET M$=N$
3040 IF N$="1" THEN LET N$="2":
GO TO 3060
3050 LET N$="1"
3060 ERASE "M";1;"NUMBER.1"
3070 OPEN #4;"M";1;"NUMBER.1"
3080 PRINT #4;N$
3090 CLOSE #4
3100 LET W$="T.BASE"+N$
3110 OPEN #4;"M";1;Y$
3120 OPEN #5;"M";1;W$
3130 INPUT #4;S$
3140 IF S$(1 TO 3)="EOF" THEN G
O TO 3190
3150 INPUT #4;T$
3160 PRINT #5;S$
3170 PRINT #5;T$
3180 GO TO 3130
3190 FOR J=1 TO S-1
3200 PRINT #5;E$(J)
3210 PRINT #5;F$(J)
3220 NEXT J
3230 PRINT #5;"EOF"
3240 CLOSE #4: CLOSE £5
3250 ERASE "M";1;Y$
3260 STOP
```

## Instructions

Now that both programs have been entered, checked and saved on cartridge, we are ready to make use of the ENGLISH/FRENCH data base.

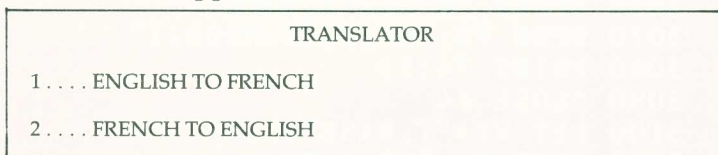
### Stage 1

First load TRANS.1 using the standard micro drive load command and then type RUN. The title INITIALIZATION will appear on the screen and the 20 English and French words contained within data lines in the program will be transferred onto the cartridge and stored in the file T.BASE.

When this operation is completed the data base is formed and we can proceed to use it with the program TRANS.2.

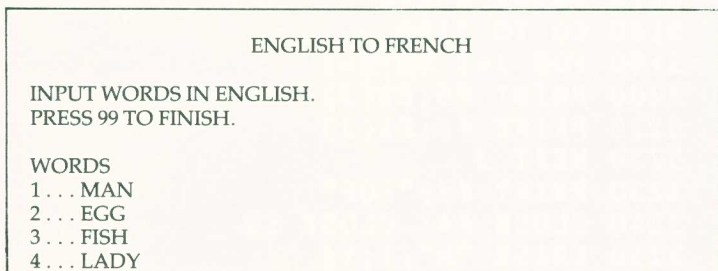
### Stage 2

Load the second program TRANS.2 and type RUN. The micro drive will operate for a short time and then the menu shown in Fig. 15 below will appear on the screen.



*Fig. 15.*

By typing 1 or 2 (anything else will be ignored) you will have the facility of an English/French or French/English dictionary. If the number 1 is entered, the user will then be required to input up to 20 English words (see Fig. 16).



*Fig. 16.*

On completion of this section (i.e. when 99 is entered), the data base will be located and the required information will be extracted and displayed on the screen using the format shown in Fig. 17.

```
ENGLISH TO FRENCH

WORD=MAN
TRANSLATION=HOMME

WORD=EGG
TRANSLATION=OEUF

WORD=LADY
TRANSLATION=DAME
```

Fig. 17.

By comparing the printouts in Fig. 16 and Fig. 17, it can be seen that the word 'fish' and its translation are missing. This is because 'fish' is not yet contained in the data base. Under these circumstances, the user is requested to locate the translation from some other source and then enter it. In this way, the data base is updated and becomes more efficient (Fig. 18).

```
ENGLISH TO FRENCH

INFORMATION NOT AVAILABLE ON THE
FOLLOWING. PLEASE RESEARCH AND
INPUT TRANSLATIONS WHEREVER
POSSIBLE. PRESS ENTER IF NO
TRANSLATION IS AVAILABLE.

WORD=FISH
TRANSLATION=POISSON

SAVING FILE
```

Fig. 18.

### Note

The initial data base is very small, but after much use it becomes more comprehensive and of greater value.

# The Network

A network is a method by which several computers can be linked together enabling data and programs to pass from one to another. The ZX-Interface.1 allows up to 64 Spectrum computers to be linked together; they can then communicate at a speed of about 3K per second. This is much faster than the communication along an RS232 connection, which has a maximum speed of 2.4K per second.

A network system can be utilized in numerous ways, ranging from education to multiuser games. It could also be used for program development, giving the equivalent of a 96K computer (with two machines linked together). Another use for a network is for several machines to share peripherals, such as printers, micro drives, etc., thus minimizing the cost when they are used in group, class or club environments.

## Establishing a Network

It is important to note that the Spectrum network uses an open-ended system, not a closed-loop or ring system as found with many other computers.

The first stage in setting up a communications network is to use the cables supplied with the interface units as shown in Fig. 19.

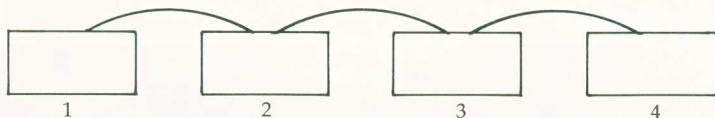


Fig. 19.

### Note

The machines located at the ends of the network have only one connection.

When the computers have been connected in the required manner, each one must be assigned its station number, using the command

```
FORMAT "N";s
```

where N is the channel specifier for the network and s is the station number in the range 1-64. After this, all stations become operational and communication can commence.

## Programs

Consider the situation where a network has been created as shown in Fig. 19 with each computer having been numbered using the relevant FORMAT command.

If the user on machine.1 wishes to send a program to machine.2, then the following steps should be taken.

- 1) User on 2 enters

```
LOAD * "n";1
```

- 2) User on 1 enters

```
SAVE * "n";2
```

When both steps have been completed, the borders of both machines will flash and the program will be transmitted.

### Note

The commands MERGE \* "n" and VERIFY \* "n" can also be used when sending and receiving programs through the network, as can the variations using CODE, DATA and SCREEN\$.

## Data Files

Consider the same network of machines and a data file called TEST containing a series of names and numbers which are to be transferred from station.1 to station.2. Provided that both station



numbers have been correctly assigned, this could be achieved using the commands

```
OPEN#N;"n";x
PRINT#x
INPUT#x
```

as indicated in the two routines shown below.

### Receiver

```
1000 OPEN#4;"n";1
1010 LET I=1
1020 INPUT#4;A$(I)
1030 IFA$(I,1 TO 3)="EOF" THEN GOTO 1070
1040 INPUT#4;A(I)
1050 LET I=I+1
1060 GOTO 1020
1070 CLOSE#4
```

### Transmitter

```
1000 OPEN#4;"n";2
1010 FOR J=1 TO I
1020 PRINT#4;A$(J)
1030 PRINT#4;A(J)
1040 NEXT J
1050 PRINT#4;"EOF"
1060 CLOSE#4
```

### Note

Careful inspection of these examples reveals the following points.

- 1) The routines are very similar to those used when sending or receiving a data file from a micro drive.
- 2) The name of the file is not stipulated on either machine and is unnecessary since there is only one file being considered. A search does not have to be undertaken as on a micro drive.
- 3) In the OPEN# statement, no mention is made of whether the file is a read or write file. This decision is made automatically by the computer when it encounters the first PRINT# or INPUT# statement.

## Synchronization

Although synchronization is not a necessity for the operation of a network system, it is preferable from a user's point of view for both programs to reach the communications routine at about the same time, thus reducing the period for which the CPU remains idle.

The following routine shows how we can make use of the INKEY\$# command to test if a network stream is ready for communication, and if not then to return to the main body of the program for further processing.

```

1000 OPEN#4;"n";1
1010 LET A$=INKEY$#4
1020 IF A$="" THEN CLOSE#4:RETURN
1030 LET I=1
1040 ETC.ETC.ETC.

```

For this routine to function the transmitting station must send out a 'flag' or 'dummy' value before the actual data file. For example

```

1000 OPEN#4;"n";1
1010 PRINT#4;"FLAG"
1020 FOR J=1 TO I
1030 PRINT#4;A$(J)
1040 PRINT#4;A(J)
1050 NEXT J
1060 PRINT#4;"EOF"
1070 CLOSE#4

```

## Broadcasting

When data is transmitted from one station to another, the principle of *handshaking* is employed. If the receiving unit is not able to accept the data, then the transmitting station will delay until the receiver is ready so that no data will be lost.

On occasions, however, we may find that a particular machine wishes to transmit a program or data file to be received by all stations that are listening. This can be achieved by using station number 0 and is known as broadcasting. 0 is used by both the

transmitting and receiving stations by typing SAVE\*"n";0 and LOAD\*"n";0, respectively.

## Adventure

The ADVENTURE program listed here is designed to be entertaining as well as informative. It also demonstrates the use of a network with a multiuser game.

### The scenario

The game is designed for up to 63 players who are given the task of finding the Golden Sceptre and returning it to its rightful place in the main chamber.

At the commencement of the game, all the adventurers are placed at the same location from which they are free to roam at will. During their many travels, they are likely to find all manner of items and weapons which can be collected and used throughout the quest, although no adventurer may carry more than five items at any time. During their meanderings, the adventurers are also likely to have encounters with monsters, which they may attack, or their fellow players, who, like themselves, are trying desperately to be the first to complete the quest. If at any time during the game a character is killed, then his possessions are left at that place and he no longer plays a part in the game, although the adventure will continue for all the other players. Play continues until (1) the quest is completed or (2) all the players are killed.

### Listing

```

1 BORDER 0: PAPER 0: INK 7: C
LS
2 INPUT "How many players (2-
63) ";no
3 IF no<2 OR no>63 THEN GO T
O 2
4 PRINT AT 5,0;"Tell the othe
r player(s) to type"
5 PRINT AT 7,9;" FORMAT ""n""
;"; FLASH 1;"?"

```

```

6 PRINT "'Where '?' is their
player number"
7 PRINT "'This computer is pl
ayer no.1's"
8 PRINT "'N.B. Player's turns
will be in"
9 PRINT 'TAB 11;"this order"
10 PRINT "'Press any key when
they have"'TAB 11;"done this"
11 PAUSE 0
12 CLS
13 PRINT AT 5,4;"Now tell them
all to type"
14 PRINT AT 7,1;" LOAD *""n"";
0: LOAD *""n"";OCODE "
15 PRINT "'And press any key w
hen they have"
16 PRINT 'TAB 11;"finished"
17 PAUSE 0
18 CLS
19 DIM s$(no,10)
20 FOR n=1 TO no
21 INPUT "Enter player no.";(n
);"s name:"; LINE s$(n)
22 NEXT n
23 DEF FN a( )=PEEK 23749
24 DIM a$(93+no*7)
25 RESTORE 9990: FOR n=1 TO 20
: READ v: LET a$(n)=CHR$ (v+(v=0
)*(10*RND+1)): NEXT n
26 FOR n=21 TO 60 STEP 2: LET
a$(n)=CHR$ (RND*20+1): LET a$(n+
1)=CHR$ (RND*70+30): NEXT n
27 LET a$(61)=CHR$ 0
28 FOR n=94 TO LEN a$ STEP 7:
LET a$(n)=CHR$ 1: LET a$(n+1)=CH
R$ 50: FOR m=n+2 TO n+6: LET a$(
m)=CHR$ 0: NEXT m: NEXT n
29 DEF FN b$(a,b,c,d)=CHR$ a+C
HR$ b+CHR$ c+CHR$ d
30 RESTORE 9900: DIM p$(20,36)
: FOR n=1 TO 20: READ p$(n): NEX
T n
31 RESTORE 9940: DIM o$(20,11)

```

```

: FOR n=1 TO 20: READ o$(n): NEX
T n
  32 DEF FN p()=CODE a$(94+(FN a
  ()-1)*7)
  33 DEF FN d(n)=CODE p$(FN p(),
  n)
  34 DEF FN d$(n)=("north" AND n
  =1)+("east" AND n=2)+("south" AN
  D n=3)+("west" AND n=4)
  35 DEF FN o$(n)=("a" AND o$(n,
  1)="a")+("an" AND o$(n,1)="n")+
  ("some" AND o$(n,1)="s")
  36 DEF FN s(n)=CODE a$(94+(n-1
  )*7)
  37 RESTORE 9950: DIM m$(20,10)
: FOR n=1 TO 20: READ m$(n): NEX
T n
  38 DEF FN z()=94+7*(FN a()-1)
  39 LET ch=0: RESTORE 40: FOR n
  =64913 TO 64918: READ v: LET ch=
  ch+v: POKE n,v: NEXT n
  40 DATA 33,140,92,54,255,201
  41 RESTORE 42: FOR n=65129 TO
  65176: READ v: LET ch=ch+v: POKE
  n,v: NEXT n
  42 DATA 255,243,197,213,229,24
  5,1,254,127,237,120,254,252,40,9
  ,205,144,253,241,225,209,193,251
  ,201,62,56,50,141,92,205,107,13,
  1,254,127,237,120,254,127,237,12
  0,254,252,40,247,251,207,20
  43 RESTORE 44: FOR n=65180 TO
  65186: READ v: LET ch=ch+v: POKE
  n,v: NEXT n
  44 DATA 62,9,237,71,237,94,201
  45 RESTORE 46: FOR n=65190 TO
  65196: READ v: LET ch=ch+v: POKE
  n,v: NEXT n
  46 DATA 62,62,237,86,237,71,20
  1
  47 IF ch<>10714 THEN CLS : PR
  INT FLASH 1;AT 10,3;"ERROR IN M
  ACHINE CODE DATA";AT 12,2;"PLEAS
  E CHECK DATA STATEMENTS";AT 14,1
  0;"LINES 40-46": STOP

```

```

48 DEF FN r(n)=CODE a$(94+(n-1
)*7+1)
98 SAVE *"n";0 LINE 100
99 SAVE *"n";0CODE 64913,285
100 LET zzz=USR 65180: FOR z=1
TO no
105 LET rr=0: FOR r=1 TO no: LE
T rr=rr+(FN r(r)<>0): NEXT r: IF
rr=0 THEN GO TO 8500
110 IF FN a(=z THEN GO SUB 10
00: GO SUB 700
120 IF FN a(<>z THEN GO SUB 5
00
130 NEXT z
140 GO TO 100
500 LET a$(62 TO 93)="Nothing i
mportant happens"
505 OPEN #4;"n";0
510 INPUT #4;a$
520 CLOSE #4
530 PRINT 'a$(62 TO 93)
540 RETURN
700 OPEN #4;"n";0
710 PRINT #4;a$
720 CLOSE #4
730 RETURN
1000 IF a$(FN z()+1)=CHR$ 0 THEN
RETURN
1002 LET o=3: GO SUB 1610: IF FN
p(=1 AND q THEN LET a$(61)=CH
R$ FN a(): GO TO 8000
1003 IF CODE a$(61) THEN PRINT
's$(CODE a$(61));" has retrieved
the""golden sceptre": INK 6:
PAPER 2: GO TO 8510
1005 PRINT '"=====
===== ""It's your turn
";s$(FN a())
1010 GO SUB 9000
1015 FOR w=21 TO 60 STEP 2: IF C
ODE a$(w)=FN p() AND RND<.5 THEN
LET n=w: GO SUB 5000
1017 IF a$(FN z()+1)=CHR$ 0 THEN
RETURN
1018 NEXT w

```

```

1020 PRINT "What do you want to
do ?": PRINT
1030 LET q$=INKEY$
1040 IF q$<"a" OR q$>"z" THEN G
O TO 1030
1047 IF q$="p" THEN RETURN
1050 IF q$="n" THEN LET d=1: GO
SUB 1500: RETURN
1055 IF q$="f" THEN GO SUB 3500
: RETURN
1056 IF q$="z" THEN GO SUB 3700
: RETURN
1057 IF q$="r" THEN GO SUB 3800
: RETURN
1060 IF q$="e" THEN LET d=2: GO
SUB 1500: RETURN
1070 IF q$="s" THEN LET d=3: GO
SUB 1500: RETURN
1080 IF q$="w" THEN LET d=4: GO
SUB 1500: RETURN
1090 IF q$="t" OR q$="g" THEN G
O SUB 2000: RETURN
1095 IF q$="l" OR q$="i" THEN G
O SUB 3000: RETURN
1100 IF q$="d" THEN GO SUB 2500
: RETURN
1110 IF q$="a" THEN GO SUB 4000
: RETURN
1480 GO TO 1030
1500 GO TO 1600
1502 IF NOT FN d(d) THEN PRINT
"The way to the ";FN d$(d);" is
blocked": RETURN
1505 LET a$(FN z())=CHR$ FN d(d)
1510 PRINT "You go to the ";FN
d$(d)
1520 LET a$(62 TO 93)=s$(FN a())
+" goes "+FN d$(d)
1530 RETURN
1600 LET p=FN p()
1605 GO TO 1620
1610 LET q=0: LET a=FN z()+2: FO
R y=a TO a+4: LET q=q+(CODE a$(y)
)=o): NEXT y

```

```

1615 RETURN
1620 RESTORE 1690: FOR o=5 TO 10
1630 READ pp,dd
1640 GO SUB 1610
1642 IF q=0 AND p=pp AND d=dd TH
EN GO TO 1699
1645 IF q<>0 AND p=pp AND d=dd T
HEN GO TO 1502
1650 NEXT o
1660 GO TO 1502
1690 DATA 4,3,4,4,7,3,12,3,16,2,
15,3
1699 PRINT '"You cannot go that
way yet !!": RETURN
2000 GO TO 2200
2005 LET q=0: PRINT '"Choices:":
PRINT
2010 FOR n=1 TO 20
2020 IF CODE a$(n)=FN p() THEN
PRINT n;" ";FN o$(n);" ";o$(n,2
TO ): LET q=1
2030 NEXT n
2040 IF q=0 THEN PRINT '"0) Not
hing": PRINT '"There is nothing
here to take": RETURN
2050 PRINT '"Which do you wish t
o take ?"
2060 INPUT "Enter choice:";o
2070 IF o<1 OR o>20 THEN GO TO
2060
2080 IF CODE a$(o)<>FN p() THEN
PRINT '"That is not here": GO T
O 2060
2090 PRINT '"You take ";FN o$(o)
;" ";o$(o,2 TO )
2095 IF o=3 THEN CLS : FOR y=1
TO 16: PRINT AT 10,6+y; FLASH 1;
PAPER RND*3; INK RND*3+4;"CONGR
ATULATIONS!"(y): BEEP .008,RND*4
0: NEXT y: PAUSE 50: PRINT '"You
have found the sceptre"' "Now a
ll you've got to do is"' "take i
t back": PAUSE 200
2100 LET a$(62 TO 93)=s$(FN a())

```



```

+" takes "+FN o$(o)+" "+o$(o,2 T
O )
2110 LET a$(o)=CHR$ O
2120 LET a$(w)=CHR$ o
2130 RETURN
2200 LET a=FN z()+2: LET qq=0
2210 FOR n=a TO a+4
2220 IF a$(n)=CHR$ O THEN LET w
=n: LET qq=1
2230 NEXT n
2240 IF qq=0 THEN PRINT '"You c
annot carry any more": RETURN
2250 GO TO 2005
2500 LET q=0: PRINT '"Choices:":
PRINT
2510 LET a=FN z()+2
2520 FOR n=a TO a+4
2530 IF a$(n)<>CHR$ O THEN PRIN
T CODE a$(n);" ";FN o$(CODE a$(
n));" ";o$(CODE a$(n),2 TO ): LE
T q=1
2540 NEXT n
2550 IF q=0 THEN PRINT "O) Noth
ing": PRINT '"You have nothing t
o drop": RETURN
2555 INPUT "Enter Choice ";o: IF
o<1 OR o>20 THEN GO TO 2555
2557 GO TO 2700
2560 LET a$(CODE a$(n))=CHR$ FN
p()
2570 PRINT '"You drop ";FN o$(CO
DE a$(n));" ";o$(CODE a$(n),2 TO
)
2580 LET a$(62 TO 93)=s$(FN a())
+" drops "+FN o$(CODE a$(n))+ " "
+o$(CODE a$(n),2 TO )
2590 LET a$(n)=CHR$ O
2600 RETURN
2700 FOR n=a TO a+4: IF CODE a$(
n)=o THEN GO TO 2560
2710 NEXT n
2720 PRINT '"You have not got th
at": RETURN
3000 LET q=0: LET a=FN z()+2: PR
INT '"You are carrying :": PRINT

```

```

3010 FOR n=a TO a+4
3020 IF a$(n)<>CHR$ 0 THEN PRIN
T FN o$(CODE a$(n));" ";o$(CODE
a$(n),2 TO ): LET q=1
3030 NEXT n
3040 IF q=0 THEN PRINT "nothing
"
3050 RETURN
3500 LET a=FN z()+2
3510 LET q=0: FOR n=a TO a+4: IF
a$(n)=CHR$ 1 OR a$(n)=CHR$ 2 OR
a$(n)=CHR$ 15 OR a$(n)=CHR$ 16
OR a$(n)=CHR$ 17 THEN LET q=n
3520 NEXT n
3530 IF q=0 THEN PRINT '"You ha
ve no food": RETURN
3540 PRINT '"You eat and become
stronger"
3545 LET a$(62 TO 93)=s$(FN a())
+" eats some food"
3550 LET a$(q)=CHR$ 0
3560 LET a$(FN z()+1)=CHR$ (CODE
a$(FN z()+1)+20)
3570 RETURN
3700 LET a=FN z()+2
3710 LET q=0: FOR n=a TO a+4: IF
a$(n)=CHR$ 4 OR a$(n)=CHR$ 18 O
R a$(n)=CHR$ 19 OR a$(n)=CHR$ 20
THEN LET q=n
3720 NEXT n
3730 IF q=0 THEN PRINT '"You ha
ve no wine": RETURN
3740 PRINT '"You drink and becom
e stronger"
3745 LET a$(62 TO 93)=s$(FN a())
+" has a drink"
3750 LET a$(q)=CHR$ 0
3760 LET a$(FN z()+1)=CHR$ (CODE
a$(FN z()+1)+10)
3770 RETURN
3800 PRINT '"You rest and recove
r some strength"
3810 LET a$(FN z()+1)=CHR$ (CODE
a$(FN z()+1)+2)
3820 RETURN

```

```

4000 PRINT "Choices:": PRINT :
LET q=0
4010 FOR i=21 TO 60 STEP 2
4020 IF CODE a$(i)=FN p() THEN
LET q=1: PRINT (i-19)/2;" the "
;m$(i-19)/2)
4030 NEXT i
4040 IF q=0 THEN PRINT "0) Nothing": PRINT "There are no monsters here": RETURN
4050 INPUT "Enter choice ";m
4060 IF m<1 OR m>20 THEN GO TO 4050
4070 IF CODE a$((m*2)+19)<>FN p() THEN PRINT "He is not here": RETURN
4080 LET n=m
4090 PRINT "You attack the ";m$(n)
4500 LET nn=(n*2)+19
4510 LET mhp=CODE a$(nn+1)
4515 PAUSE 50
4520 LET hp=CODE a$(FN z()+1)
4525 LET r=RND
4530 IF r>.5 AND mhp>hp THEN BE EP .3,20: LET a$(62 TO 93)="The "+m$(n)+" hits "+s$(FN a()): PRINT "You are hit by the ";m$(n): LET hp=hp-((1/5)*mhp): IF hp<=0 THEN GO TO 4700
4535 IF r<=.5 AND mhp>hp THEN PRINT "You are missed by the ";m$(n): LET a$(62 TO 93)=s$(FN a()+1)+" is lucky": RETURN
4540 IF mhp=hp THEN PRINT "The fight is drawn": LET a$(62 TO 93)="There is a fight, but it is drawn": RETURN
4550 IF r>.5 AND mhp<hp THEN BE EP .3,0: LET a$(62 TO 93)=s$(FN a()+1)+" hits the "+m$(n): PRINT "You hit the ";m$(n): LET mhp=mhp-((1/5)*hp): IF mhp<=0 THEN GO TO 4800

```

```

4555 IF r<=.5 AND mhp<hp THEN P
RINT "You miss the ";m$(n): LET
a$(62 TO 93)=s$(FN a())+" misse
s the "+m$(n): RETURN
4560 LET a$(nn+1)=CHR$ mhp
4570 LET a$(FN z()+1)=CHR$ hp
4580 RETURN
4700 PRINT "You are dead"
4710 LET a$(62 TO 93)=s$(FN a())
+" is dead"
4720 LET a=FN z()+2
4730 FOR n=a TO a+4
4740 IF CODE a$(n)<>0 THEN LET
a$(CODE a$(n))=CHR$ FN p()
4750 LET a$(a-1)=CHR$ 0
4760 RETURN
4800 PRINT "The monster is dead
"
4810 LET a$(62 TO 93)=s$(FN a())
+" kills the "+m$(n)
4820 LET a$(nn)=CHR$ 0
4830 RETURN
5000 LET n=(n-19)/2
5010 PRINT "You are attacked by
the ";m$(n)
5020 GO TO 4500
8000 PAPER 6: INK 2: BORDER 6: C
LS
8010 PRINT "You have completed
your quest"
8020 PRINT "You are a super-int
elligent"" mega-being"
8025 PAUSE 100
8030 CLS
8040 POKE 23659,0
8050 PRINT "Congratulations ";
8060 BEEP .08,RND*50
8070 GO TO 8050
8400 STOP
8500 BORDER 2: PAPER 2: INK 6: C
LS : PRINT "You are all dead""
The game has finished"
8510 FOR n=0 TO 50: BEEP .005,n:
BEEP .005,50-n: NEXT n

```

```

8520 FLASH 1: CLS
8530 PRINT AT 10,10; INVERSE 1;"
HARD LUCK!"
8535 POKE 23659,0
8540 BORDER 6: BORDER 2: GO TO 8
540
8600 STOP
9000 PRINT '"You are ";p$(FN p(
,5 TO )
9010 PRINT '"Visible exits are :
"

9015 PRINT
9020 FOR n=1 TO 4
9030 IF FN d(n)<>0 THEN PRINT F
N d$(n)
9040 NEXT n
9050 PRINT '"You see :"'
9055 PRINT : LET q=0
9060 FOR n=1 TO 20
9070 IF CODE a$(n)=FN p() THEN
PRINT FN o$(n);" ";o$(n,2 TO ):
LET q=1
9080 NEXT n
9090 FOR n=1 TO no
9100 IF FN s(n)=FN p() AND n<>FN
a() THEN PRINT s$(n): LET q=1
9110 NEXT n
9120 FOR n=21 TO 60 STEP 2
9130 IF CODE a$(n)=FN p() THEN
PRINT "the ";m$((n-19)/2): LET q
=1
9140 NEXT n
9150 IF q=0 THEN PRINT "nothing
"

9160 RETURN
9900 DATA FN b$(0,0,2,0)+"in the
main cavern"
9901 DATA FN b$(1,3,5,4)+"at the
parting of the ways"
9902 DATA FN b$(0,0,6,2)+"in a s
mall cave"
9903 DATA FN b$(0,2,9,7)+"next t
o a stinking well"
9904 DATA FN b$(2,0,0,9)+"in the
secret passage"

```

9905 DATA FN b\$(0,3,13,14)+"in t  
 he giant's dark dungeons"  
 9906 DATA FN b\$(0,4,8,1)+"on a s  
 tone ledge"  
 9907 DATA FN b\$(7,0,12,11)+"on t  
 he edge of a precipice"  
 9908 DATA FN b\$(4,5,10,0)+"in th  
 e troll's cave"  
 9909 DATA FN b\$(9,13,0,0)+"by th  
 e spider's web"  
 9910 DATA FN b\$(0,8,17,0)+"in an  
 open space"  
 9911 DATA FN b\$(8,0,15,0)+"in a  
 misty marsh"  
 9912 DATA FN b\$(0,0,14,6)+"in a  
 narrow passage"  
 9913 DATA FN b\$(13,1,1,6)+"under  
 a waterfall"  
 9914 DATA FN b\$(12,10,16,0)+"in  
 a smelly bog"  
 9915 DATA FN b\$(15,20,19,0)+"on  
 a winding staircase"  
 9916 DATA FN b\$(11,18,19,18)+"in  
 a dank old storeroom"  
 9917 DATA FN b\$(8,17,0,0)+"at th  
 e bottom of a sheer cliff"  
 9918 DATA FN b\$(17,0,0,0)+"by a  
 tiny black stream"  
 9919 DATA FN b\$(0,0,0,16)+"in TH  
 E GREAT CHAMBER"  
 9940 DATA "sfood","sfood","ascep  
 tre","swine","naxe","srope","ash  
 ield","akey","ahelmet","asword",  
 "akey","naxe","acloak","awand",  
 "sfood","sfood","sfood","swine",  
 "swine","swine"  
 9950 DATA "goblin","wolf","ghoul  
 ","dragon","giant","spider","bea  
 st","bat","giant ant","vampire",  
 "ghost","troll","scorpion","snak  
 e","gorgon","medusa","serpent",  
 "dinosaur","ape","goblin","spider  
 "  
 9990 DATA 0,0,20,0,14,13,10,18,1  
 9,12,0,0,0,0,0,0,0,0,0,0

## Instructions

All the instructions required to set up and play the adventure are contained within the program and these should be followed very carefully. During the game all decisions are made using a single keystroke, as listed below.

e	East
n	North
s	South
w	West
g	Get object
d	Drop object
p	Pass
l	List possessions
a	Attack monster
z	Drink
f	Food
r	Rest

# The RS232 Interface

The ZX-Interface.1 also incorporates an RS232 interface, which can be used for serial communication with various peripheral devices, such as printers, terminals and modems. The rate at which this communication takes place will depend on the peripheral being used and can be set, using the keyboard, to any of the standard baud rates listed below.

Baud rates	
50	100
300	600
1200	2400
4800	9600
19200	

1 baud = 1 bit per second

## Note

It is important not to get confused between bits and bytes (remember that 1 byte = 8 bits). Therefore to transmit 16K at 600 baud would take  $1024 \times 8 \times 16/600$  seconds (i.e. 218.45 seconds). As well as different baud rates, most peripherals will require the data in slightly different formats. The ZX-Interface.1 uses: 8 bits; no parity; 1 stop bit. When connecting any peripheral to the interface, these should be taken into account and the necessary adjustments made.

## Commands

Fig. 20 displays the commands that can be used in connection with the RS232 interface.



```
FORMAT "t"; BAUD
FORMAT "b"; BAUD
OPEN#3;"t"
OPEN#n;"b"
PRINT#n
LPRINT
LLIST
```

*Fig. 20.*

Let us now look at each of the above terms in more detail, considering the method of entry, their use and any variations that can be employed.

### **FORMAT "t"; BAUD/FORMAT "b"; BAUD**

Before the RS232 interface can be used, the transmission rate must be specified by using the FORMAT command with either t or b as the channel specifier.

### **OPEN#3;"t"**

This opens stream 3 for the t channel of the RS232 interface. We could use any stream number (3–15), but it is preferable to use #3 for reasons indicated later.

### **OPEN#n;"b"**

This opens stream n for the b channel of the RS232 network. In this case, there is no specific advantage in using any particular stream number.

### **PRINT#n**

When the baud rate has been specified and a stream has been opened, information can be sent to a peripheral by using the PRINT# command.

## LPRINT

If stream #3 has been opened for the t channel, then information can be sent to a peripheral using the LPRINT command instead of PRINT#3. This reduces the number of keystrokes required to enter a line and also occupies less memory.

## LLIST

This is shorthand for LIST#3, saving on keystrokes and memory space.

## The Two Channels

From the previous pages we observe that the RS232 interface uses two channels, specified by t and b.

### *The t Channel*

The t or text channel is generally used for sending listings to a printer or for producing a hard copy during the execution of a program. The main factor of this mode is that certain control codes are ignored and not sent to the printer.

Characters 0–12	Control codes ignored.
Characters 13	Carriage return followed by a new line automatically sent.
Characters 14–31	Control codes ignored.
Characters 32–127	ASCII codes sent as normal.
Characters 128–164	Graphics characters sent as “?” (CODE 63).
Characters 165–255	Keywords expanded into ASCII form.

### *The b Channel*

The b or binary channel is so named since it sends the full 8-bit code as used by the Spectrum. This is very useful because it enables us to send control codes to the peripherals.

## Control Codes

Consider the Tandy four-colour plotter–printer used in the production of parts of this book. To change the device from text

into graphics mode the code CHR\$18 is required. If we were to use the routine

```
10 FORMAT"t";600
20 OPEN#1;"t"
30 PRINT#3;CHR$18
```

the CHR\$18 would be ignored since the code lies in the range 14–31 (control codes).

To overcome this problem, we would have to make use of the b channel

```
10 FORMAT"b";600
20 OPEN#4;"b"
30 PRINT#4;CHR$18
```

In this case, the correct code would be sent.

### Example

The following program uses the Tandy four-colour printer-plotter to produce a *pie-chart*. It demonstrates how both the t and the b channels can be used simultaneously within a program.

#### *Listing*

```
10 DIM N$(10,10)
20 DIM U(10)
30 DIM P(10)
40 DIM D(100)
50 DIM C(10)
60 DIM C$(4)
70 LET L=0
80 CLS
90 PRINT TAB (10);"PIE-CHART"
100 PRINT TAB (10);"=== ====="
110 INPUT "TITLE=";T$
120 INPUT "RADIUS=";R
130 INPUT "NO.OF ITEMS=";N
140 LET M=0
150 LET A$=CHR$ 10
160 GO SUB 1090
170 FORMAT "b";600
180 LET CC=0
```

```

190 FORMAT "t";600
200 OPEN #3;"t"
210 OPEN #4;"b"
220 LPRINT T$
230 PRINT #4;A$
240 PRINT #4;CHR$ 18
250 LPRINT "MO,-120,120,-120"
260 LPRINT "I"
270 GO SUB 460
280 LPRINT "M130,0,130,";(N-1)*
15
290 FOR I=1 TO N
300 LPRINT "I"
310 LPRINT "C";C(I)
320 LPRINT "JO,20,30,0,0,-20,-3
0,0"
330 LET J=0
340 LPRINT "MO,";J: LPRINT "J30
,0"
350 LET J=J+P(I): IF J<20 THEN
GO TO 340
360 LPRINT "M40,0": LPRINT "P";
N$(I)
370 FOR K=1 TO 11-LEN (N$(I))
380 LPRINT "P": NEXT K
390 LPRINT "P";U(I)
400 LPRINT "MO,0,0,-30"
410 NEXT I
420 LPRINT "MO,-100": LPRINT "A
"

430 CLOSE #3
440 CLOSE #4
450 STOP
460 LET PI=3.1416: LET P2=2*PI
470 LET A$="M"
480 FOR I=0 TO 100 STEP 2
490 LET S=I/100*P2
500 LET X=INT (SIN (S)*R): LET
Y=INT (COS (S)*R)
510 LPRINT A$;X;",";Y: LET A$="
D"
520 NEXT I
530 LET S=0
540 FOR I=1 TO N

```

```

550 LET S=S+U(I)*P2/M
560 LET X=INT (SIN (S)*R): LET
Y=INT (COS (S)*R)
570 LPRINT "HD";X;",";Y
580 NEXT I
590 LET P=0
600 FOR I=1 TO N
610 LPRINT "C";C(I)
620 LET Q=P: LET P=P+U(I)
630 LET QS=Q*P2/M: LET PS=P*P2/
M
640 LET X1=INT (SIN (QS)*R): LE
T QY=INT (COS (QS)*R)
650 LET X0=INT (SIN (PS)*R): LE
T PY=INT (COS (PS)*R)
660 LET ST=R: LET EN=-R
670 IF QY>=0 AND PY>=0 THEN LE
T EN=0
680 IF QY<=0 AND PY<=0 THEN LE
T ST=0
690 FOR Y=ST TO EN STEP -P(I)
700 LET J=1
710 LET RA=SQR (R*R-Y*Y)
720 IF Y=0 THEN LET RS=PI/2: G
O TO 0750
730 LET RS=ATN (RA/Y)
740 IF RS<0 THEN LET RS=RS+PI
750 IF QS<RS AND RS<PS THEN LE
T D(J)=INT (RA): LET J=J+1
760 LET RS=P2-RS
770 IF QY=0 THEN GO TO 0830
780 LET X=X1/QY*Y
790 IF SGN (X)<>SGN (X1) THEN
GO TO 0830
800 IF SGN (Y)<>0 AND SGN (QY)<
>SGN (Y) THEN GO TO 0830
810 LET RT=SQR (X*X+Y*Y)
820 IF RT<=R THEN LET D(J)=INT
(X): LET J=J+1
830 IF PY=0 THEN GO TO 0890
840 LET X=X0*Y/PY
850 IF SGN (X)<>SGN (X0) THEN
GO TO 0890
860 IF SGN (Y)<>0 AND SGN (PY)<
>SGN (Y) THEN GO TO 0890

```

```

870 LET RT=SQR (X*X+Y*Y)
880 IF RT<=R THEN LET D(J)=INT
(X): LET J=J+1
890 IF QS<RS AND RS<PS THEN LE
T D(J)=INT (-RA): LET J=J+1
900 IF Y=0 AND J<>3 THEN LET D
(J)=0: LET J=J+1
910 IF J<=3 THEN GO TO 0990
920 FOR K=1 TO J-1
930 LET MIN=D(K): LET MN=L
940 FOR L=K+1 TO J-1
950 IF D(L)<MIN THEN LET MIN=D
(L): LET MN=L
960 NEXT L
970 LET D(MN)=D(K): LET D(K)=MI
N
980 NEXT K
990 LET K=1
1000 IF J=2 AND Y=0 AND CC=1 THE
N LET D(K)=-R: LET D(K+1)=0: LE
T CC=2
1010 IF J<=2 THEN GO TO 1060
1020 IF Y=0 AND J=4 AND CC=0 THE
N LET D(K)=0: LET D(K+1)=R: LET
CC=1
1030 LPRINT "M";D(K);",",";Y: LPRI
NT "D";D(K+1);",",";Y
1040 LET VV=J
1050 LET K=K+2: IF K<J-1 THEN G
O TO 1020
1060 NEXT Y
1070 NEXT I
1080 RETURN
1090 FOR I=1 TO N
1100 INPUT "ITEM=";N$(I)
1110 INPUT "FREQUENCY=";U(I)
1120 INPUT "COLOUR=";C$(I)
1130 IF C$(I)="BLACK" THEN LET
C(I)=0
1140 IF C$(I)="BLUE" THEN LET C
(I)=1
1150 IF C$(I)="GREEN" THEN LET
C(I)=2
1160 IF C$(I)="RED" THEN LET C(
I)=3

```

```
1170 INPUT "SHADING=";P(I)
1180 LET M=M+U(I)
1190 NEXT I
1200 RETURN
```

PIE-CHART SHOWING SALES

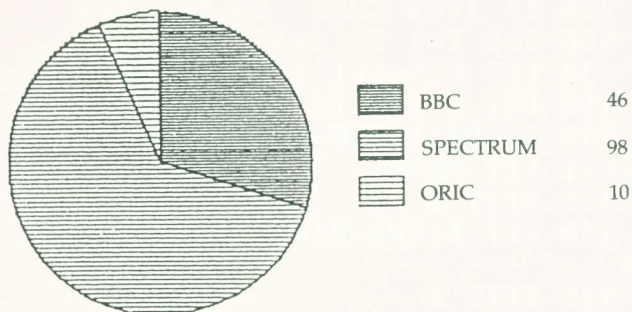
*Fig. 21.*

Fig. 21 shows an example of a printout obtained using the PIE-CHART program.

### Other Peripherals

In this chapter, we have considered how the RS232 interface can be used to connect the Spectrum to a printer. It must be realized that in this case communication is taking place in one direction only, with the computer acting as transmitter and the printer as receiver. It was mentioned in the introduction, however, that it is also possible to connect the RS232 interface to other devices, such as modems, terminals or even other computers. In these cases, the communication may well be bidirectional and we must use only the b channel, which will send or receive the full 8-bit code.

# Slide Show

Throughout this book we have considered many applications which are made available by using micro drives and the ZX-Interface.1. In this chapter we will look at one final application, made possible by the speed at which data can be transferred from a cartridge into the computer and then onto the screen. The two programs listed here — SLIDE.1 and SLIDE.2 — employ files, machine code and the SCREEN\$ function to make the computer act as a display unit for a series of pictures or screens of text. They allow for a maximum of 26 screens to be constructed, edited, saved on a cartridge and then displayed in sequence to provide the user with a display tool of great complexity.

## SLIDE.1

This short program initializes the system and generates the required files. It should be typed, checked for errors, and then saved on cartridge using the command

```
SAVE*"m";1;"SLIDE.1"
```

### *Listing*

```
10 CLS
20 LET a$=""
30 OPEN #4;"m";1;"SLIDES"
40 FOR i=1 TO 26
50 LET a$=a$+" "
60 NEXT i
70 PRINT #4;a$
80 CLOSE #4
90 STOP
```



**SLIDE.2**

The second program — SLIDE.2 — is very much longer than the first and contains all the necessary code to enable the user to draw, edit, delete or view the various screens. It should be typed into the computer and then saved using the command

```
SAVE*"m";1;"SLIDE.2"
```

*Listing*

```

1 CLS
2 CLEAR 49999
3 GO SUB 9000
10 RESTORE
20 FOR n=65000 TO 65011
30 READ m
40 POKE n,m
50 NEXT n
60 DATA 1,0,27,17,0,64,33,80,1
95,237,176,201
70 BORDER 7: INK 0: PAPER 7: F
LASH 0: BRIGHT 0: OVER 0: CLS
75 PRINT AT 1,10;"Menu"; OVER
1;AT 1,10;"  "
80 PRINT "'1) Draw and save a
screen"
90 PRINT "'2) View all screens
"
100 PRINT "'3) View and edit a
screen"
105 PRINT "'4) End"
107 PRINT "'5) Erase a screen"
108 PRINT "'6) List screens"
110 PRINT "'TAB 10;"Enter choic
e"
120 PAUSE 0
130 LET a$=INKEY$
140 IF a$<"1" OR a$>"6" THEN G
O TO 120
145 CLS
150 GO TO 500*VAL a$

```

```

500 LET w=0: LET x=0: LET y=0:
LET i=0: LET p=7: LET f=0: LET b
=0: LET o=0
510 PLOT INK i; PAPER p; FLASH
f; BRIGHT b; OVER o;x,y
515 IF w THEN PLOT OVER 1;x,y
520 LET a$=INKEY$
540 LET z=510
544 LET xx=x: LET yy=y
545 IF a$="w" THEN LET w=NOT w
550 IF a$="7" AND y<175 THEN L
ET y=y+1: GO TO z
560 IF a$="6" AND y>0 THEN LET
y=y-1: GO TO z
570 IF a$="5" AND x>0 THEN LET
x=x-1: GO TO z
580 IF a$="8" AND x<255 THEN L
ET x=x+1: GO TO z
590 IF a$="q" THEN INPUT "Bord
er=";s: BORDER s: GO TO z
600 IF a$="c" THEN CLS : GO TO
z
610 IF a$="i" THEN INPUT "Ink="
";i: GO TO z
620 IF a$="p" THEN INPUT "Pape
r=";p: GO TO z
630 IF a$="f" THEN LET f=NOT f
: GO TO z
640 IF a$="b" THEN LET b=NOT b
: GO TO z
650 IF a$="o" THEN LET o=NOT o
: GO TO z
660 IF a$="t" THEN INPUT "Text
="; LINE b$: PRINT INK i; PAPER
p; FLASH f; BRIGHT b; OVER o;AT
(175-y)/8,x/8;b$: GO TO z
670 IF a$="h" THEN INPUT "Radi
us=";r: CIRCLE INK i; PAPER p;
FLASH f; BRIGHT b; OVER o;x,y,r:
GO TO z
680 IF a$="d" THEN INPUT "How
far: x=";ax;" y=";ay;" z=";az: I
F x+ax>0 AND x+ax<255 AND y+ay>0
AND y+ay<175 THEN LET x=x+ax:

```

```

LET y=y+ay: DRAW INK i; PAPER p
; FLASH f; BRIGHT b; OVER o;ax,a
y,az: GO TO z
690 IF a$="1" AND x>0 AND y>0 T
HEN LET x=x-1: LET y=y-1: GO TO
z
700 IF a$="2" AND x>0 AND y<175
THEN LET x=x-1: LET y=y+1: GO
TO z
710 IF a$="3" AND x<255 AND y<1
75 THEN LET x=x+1: LET y=y+1: G
O TO z
720 IF a$="4" AND x<255 AND y>0
THEN LET x=x+1: LET y=y-1: GO
TO z
725 IF a$="m" THEN INPUT "Move
to: x=";x;" y=";y: GO TO z
730 IF a$="s" THEN GO TO 800
740 GO TO z
800 PRINT #0;"Insert cartridge
required": PAUSE 200: INPUT "":
PRINT #0;"Do not remove it while
the red": PAUSE 200: INPUT "":
PRINT #0;"light is on...Do not p
ress BREAK": PAUSE 200
805 INPUT "Which screen (A-Z) "
; LINE a$
810 IF a$<"A" OR a$>"Z" THEN G
O TO 805
820 IF q$(CODE a$-64)="." THEN
ERASE "m";1;a$
850 SAVE *"m";1;a$SCREEN$ : LET
q$(CODE a$-64)="." : CLS : GO TO
8000
1000 IF q$(1)="." THEN LOAD *"m
";1;"A"CODE
1005 IF q$(1)=" " THEN GO TO 10
70
1010 IF q$(2)="." THEN LOAD *"m
";1;"B"CODE 50000
1015 IF q$(2)=" " THEN PAUSE 0:
GO TO 1070
1020 FOR n=CODE "C" TO CODE "Z"
1025 PAUSE 0
1030 LET 1=USR 65000

```

```

1040 IF q$(n-64)="." THEN LOAD
*"m";1;CHR$ nCODE 50000
1050 IF q$(n-64)=" " THEN GO TO
1070
1060 NEXT n
1070 IF INKEY$<>"" THEN GO TO 1
070
1075 PAUSE 0
1080 CLS : PRINT #0;"No more scr
eens": PAUSE 200: GO TO 70
1500 INPUT "Which screen (A-Z) "
; LINE a$
1510 IF a$<"A" OR a$>"Z" THEN G
O TO 1500
1520 IF q$(CODE a$-64)="." THEN
LOAD *"m";1;a$SCREEN$
1530 GO TO 500
2000 PRINT AT 21,0;"Bye"
2010 STOP
2500 INPUT "Which screen (A-Z) "
; LINE a$
2510 IF a$<"A" OR a$>"Z" THEN G
O TO 2500
2520 IF q$(CODE a$-64)=" " THEN
PRINT #0;"No such file exists":
PAUSE 200: GO TO 2500
2530 ERASE "m";1;a$
2540 LET q$(CODE a$-64)=" "
2550 GO TO 8000
3000 PRINT "Screens that exist a
re shown in inverse:'''
3020 FOR n=CODE "A" TO CODE "Z"
3030 IF q$(n-64)=" " THEN INVER
SE 0
3040 IF q$(n-64)="." THEN INVER
SE 1
3050 PRINT CHR$ n;
3060 NEXT n
3065 INVERSE 0
3070 PRINT #0;"Press any key": P
AUSE 0
3080 GO TO 70
8000 ERASE "m";1;"SLIDES"
8010 OPEN #4;"m";1;"SLIDES"
8020 PRINT #4;q$

```

```

8030 CLOSE #4
8040 GO TO 70
9000 PRINT TAB 10;"SLIDESHOW"
9010 PRINT TAB 10;"======"
9020 PRINT ''
9030 PRINT AT 10,9; FLASH 1;"FIL
E LOADING"
9035 PRINT AT 15,9; INVERSE 1; F
LASH 1;"PLEASE WAIT!"
9040 OPEN #4;"m";1;"SLIDES"
9060 INPUT #4;q$
9070 CLOSE #4
9080 INPUT "PRESS ENTER TO CONTI
NUE."; LINE z$
9090 RETURN

```

## Instructions

Now that both programs have been entered and saved on a cartridge, we are ready to explore the possibilities afforded by these programs.

### Stage 1

Load the SLIDE.1 program and type RUN. The data file SLIDES will then be initialized to accept the data from the main program.

### Stage 2

When the file has been initialized, the main program SLIDE.2 should be loaded and executed. After this, the main menu will appear on the screen (see Fig. 22).

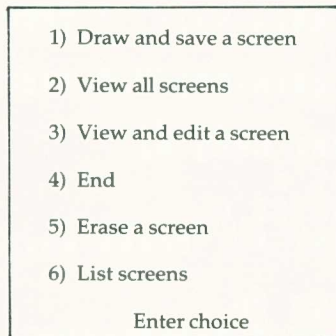


Fig. 22.

By typing a number in the range 1–6 (anything else will be ignored), access can be gained to any of the facilities listed. These are described in more detail below.

### Draw and Save a Screen

This section enables the user to design a screen of information (graphics or text) and save it on a cartridge. It is here that the user's creative ability is tested to the full, and by using the keys shown in Table 4 complex pictures can be generated.

**Table 4**

<i>Key</i>	<i>Feature</i>
1	Draw line SW
2	Draw line NW
3	Draw line NE
4	Draw line SE
5	Draw line W
6	Draw line S
7	Draw line N
8	Draw line E
w	Change WRITE status
q	Change border colour
c	Clear screen
i	Change ink colour
p	Change paper colour
f	Change FLASH status
b	Change BRIGHT status
o	Change OVER status
t	Input and print text
h	Draw circle
d	Draw an arc
m	Move to new position
s	Save screen onto cartridge

#### *Note*

Consult the Spectrum manual for further details on the meanings of the commands OVER, BRIGHT, etc.

### View all Screens

This is the 'slide show' facility in which all the screens on the cartridge are displayed in order. While the first screen is being

viewed, the next is automatically loaded into 'high memory' and then transferred to the display by a machine code routine at the appropriate time.

### **View and Edit a Screen**

This option will cause a specific screen to be displayed with the computer then entering the DRAW mode (option 1). The picture can then be edited and saved again.

### **End**

This enables us to exit from the program.

### **Erase a Screen**

As the title suggests, this option allows any screen stored on a cartridge to be erased.

### **List Screens**

The screens are saved on cartridge using the letters A-Z. By using option 6, we are able to investigate which of these names are still available for future screens.

# The ZX-Interface.2

The second extension unit for the Spectrum is unlike its predecessor because its extra facilities are limited to two areas. Firstly, it enables the user to use cartridge- or ROM-based software, and secondly, it allows twin joysticks to be used, thus alleviating the excessive use of certain keys.

It is far beyond the scope of this book to consider how the user could design his own ROM cartridges, but we shall consider the joysticks and how they may be used in various types of program.

## Commands

The joysticks are linked very closely to the keys on the top row of the keyboard, as indicated in Table 5.

Table 5

<i>Key</i>	<i>Joystick equivalent</i>
1	LEFT stick.2
2	RIGHT stick.2
3	DOWN stick.2
4	UP stick.2
5	FIRE stick.2
6	LEFT stick.1
7	RIGHT stick.1
8	DOWN stick.1
9	UP stick.1
0	FIRE stick.1

Because of the close link, pressing key.1 is exactly the same as moving stick.2 to the left. Therefore we are able to use the INKEY\$ command to check for a particular response. This



method, although satisfactory in some cases, has the major drawback that it cannot detect two functions, such as LEFT and FIRE, at the same time.

Alternatively, we could use the IN command to translate the joystick's action. By typing IN 61438, we obtain the values for stick.1 and similarly IN 63486 gives values for stick.2. The IN statement is very much like a PEEK function since it gives an 8-bit number which lies in the range 0–255. For the joysticks, we are interested only in the first five of the 8 bits as shown in Table 6.

**Table 6**

<i>Option</i>	<i>Stick.1 IN 61438</i>	<i>Stick.2 IN 63486</i>
Fire	Bit 0	Bit 4
Up	Bit 1	Bit 3
Down	Bit 2	Bit 2
Right	Bit 3	Bit 1
Left	Bit 4	Bit 0

To test for a joystick reading, we must obtain the value using IN and then extract the relevant bits as shown in the following example.

### Example

The following routine reads stick.1 and sets the variables (Up, Down, Right, Left, Fire) to 1 when an appropriate reading is taken.

```

10 CLS
20 LET UP=0
30 LET DOWN=0
40 LET RIGHT=0
50 LET LEFT=0
60 LET FIRE=0
70 LET A=255-IN 61438
80 IF A>127 THEN LET A=A-128
90 IF A>63 THEN LET A=A-64
100 IF A>31 THEN LET A=A-32

```

```

110 IF A>15 THEN LET A=A-16:LET LEFT=1
120 IF A>7 THEN LET A=A-8:LET RIGHT=1
130 IF A>3 THEN LET A=A-4:LET DOWN=1
140 IF A>1 THEN LET A=A-2:LET UP=1
150 IF A=1 THEN LET FIRE=1
160 REM AT THIS POINT THE VARIABLES ARE
    INDICATING THE JOYSTICK SITUATION

```

### Comments

Lines 10-60     Initialize all variables to zero.  
 Line 70         Take a reading.  
 Lines 80-100    Remove unwanted information stored in bits 5, 6  
                   and 7.  
 Line 110        Check bit 4, reduce and set variable.  
 Line 120        Check bit 3, reduce and set variable.  
 Line 130        Check bit 2, reduce and set variable.  
 Line 140        Check bit 1, reduce and set variable.  
 Line 150        Check bit 0 and set variable.

Now that we have seen how a joystick reading may be obtained, we will consider some applications. In the next few pages there are examples of how joysticks can be used for utilities, and complex graphical adventures.

## SKETCH

The program should be typed into the computer, checked and then saved using the command

```
SAVE*"m";1;"SKETCH"LINE 10
```

or, if you do not yet own a micro drive:

```
SAVE"SKETCH"LINE 10
```

### Listing

```

10 PAPER 0: LET ink=7: BORDER
0: CLS
15 INK 7
16 PRINT AT 17,0;"

```

```

"
20 PLOT 0,39: DRAW 255,0
30 FOR i=7 TO 169 STEP 16
40 PLOT i,16: DRAW 9,0: DRAW 0
,-9: DRAW -9,0: DRAW 0,9: NEXT i
45 PRINT AT 20,17;"R"
46 PRINT AT 20,19;"C"
47 PRINT AT 20,21;"S"
50 INK 1: LET i=8: GO SUB 1000
55 INK 2: LET i=24: GO SUB 100
0
60 INK 3: LET i=40: GO SUB 100
0
65 INK 4: LET i=56: GO SUB 100
0
70 INK 5: LET i=72: GO SUB 100
0
75 INK 6: LET i=88: GO SUB 100
0
80 INK 7: LET i=104: GO SUB 10
00
85 INK 0: LET i=120: GO SUB 10
00
90 INK 7
100 PRINT AT 18,0;"Ink Colours
:-"
110 INK ink
150 LET x=68: LET y=128
160 LET ex=x: LET why=y
200 LET a=IN 61438
210 IF a>127 THEN LET a=a-128
220 IF a>63 THEN LET a=a-64
230 IF a>31 THEN LET a=a-32
240 IF a>15 THEN LET a=a-16: I
F x<255 THEN LET ex=ex+1
250 IF a>7 THEN LET a=a-8: IF
x>0 THEN LET ex=ex-1
260 IF a>3 THEN LET a=a-4: IF
y<175 THEN LET why=why+1
270 IF a>1 THEN LET a=a-2: IF
y>0 THEN LET why=why-1
280 LET fire=a

```

```

300 IF why<39 THEN GO TO 5000
400 LET x=ex: LET y=why
410 PLOT x,y
420 IF fire=1 THEN INK 0: PLOT
x,y: INK ink
430 GO TO 200
1000 FOR x=i TO i+7: PLOT x,15:
DRAW 0,-7: NEXT x: RETURN
5000 LET x=ex: LET y=why: PLOT x
,y
5010 IF y<16 AND y>7 THEN GO TO
5030
5020 GO TO 200
5030 IF x>7 AND x<16 AND fire=0
THEN LET ink=1: GO TO 15
5035 IF x>23 AND x<32 AND fire=0
THEN LET ink=2: GO TO 15
5040 IF x>39 AND x<48 AND fire=0
THEN LET ink=3: GO TO 15
5045 IF x>55 AND x<64 AND fire=0
THEN LET ink=4: GO TO 15
5050 IF x>71 AND x<80 AND fire=0
THEN LET ink=5: GO TO 15
5055 IF x>87 AND x<96 AND fire=0
THEN LET ink=6: GO TO 15
5060 IF x>103 AND x<112 AND fire
=0 THEN LET ink=7: GO TO 15
5065 IF x>119 AND x<128 AND fire
=0 THEN LET ink=7: GO TO 15
5070 IF x>135 AND x<144 AND fire
=0 THEN GO TO 15
5075 IF x>151 AND x<160 AND fire
=0 THEN RUN
5077 IF x>169 AND x<176 AND fire
=0 THEN SAVE *"m";1;"picture"SC
REEN$: GO TO 15
5080 GO TO 200

```

## Instructions

To use the SKETCH utility, first connect the ZX-Interface.2 as shown in the instruction manual and plug in stick.1. When this has been completed, switch the Spectrum on and load the program in the normal manner. When the loading procedure is

completed the program will run automatically, and from this point on the keyboard is not used. All the commands, such as INK, CLS, SAVE, etc., are under joystick control.

### Facilities

**DRAW** Move the joystick in required direction, holding the fire button at the same time.

**MOVE** Move the joystick without holding the fire button.

**CHANGE INK** Use the joystick to move into the correct square in the palette section and press FIRE.

**CLEAR SCREEN** As above, but move into the square marked C.

**RESET PALETTE** As above, but move into the square marked R.

**SAVE SCREEN** Move into the square marked S and press FIRE.

### Note

When using the palette section of the screen, a trail will be left to help indicate your current position. This will be deleted after a selection has been made.

## MUTE HUNTER

Type the program into the computer and save it on cartridge or tape using

SAVE\*"m";1;"MUTE"

or

SAVE"MUTE"

### Listing

```

10 CLEAR
20 LET a$=""....."
....."
30 LET a$=a$+".....d..d.....
.....d          .a.. ..d ..
....."
....."

```

```

40 LET a$=a$+" . . . . .
. . . . .
. "
50 LET a$=a$+" . . . . .
. . . . . hhhh . . . . . r . . . . .
. "
60 LET a$=a$+" . .
y . . . . . . b . . . . .
. "
70 LET a$=a$+" . . . . .
. . . . . . . . . . . d . . . . .
. "
80 LET a$=a$+" . . . . .
. . . . . . . . . . .
y . "
90 LET a$=a$+" . . . . . m
. . . . . . . . . . . c . . . . .
. "
100 LET a$=a$+" . g . . . . .
. . . . . . . . . . . m . . . . .
. "
110 LET a$=a$+" . . . . . y . . . . .
. . . . . t . . . . . . . . . . .
. "
120 LET a$=a$+" . . . . .
. . . . . r . . . . . . . . . . .
y . "
130 LET a$=a$+" . l . . . . .
. i . . . . . ww . . . . .
. "
140 LET a$=a$+" . . . . . l . . . . .
. . . . . f . . . . . . . . . . . m
. "
150 LET a$=a$+" . . . . . m
. . . . . m . . . . . g . . . . .
. "
160 LET a$=a$+" . f . . . . .
. . . . . . . . . . .
. "
170 LET a$=a$+" . . . . . o . . . . .
. . . . . . . . . . .
. "
180 LET a$=a$+" . . . . . p .
. n . . . . . d . . . . .
. . "

```

```

190 LET a$=a$+".. .....
.. g . . d ..
."
200 LET a$=a$+" . . . . .
.. . . . . . . . . . .
."
210 LET a$=a$+".. .. . . .
.. . . . . . . . . . .
."
220 LET a$=a$+"... .. . ..
. ....m. .... .. .x
."
230 LET a$=a$+".. .. . . . . .
... r... . .d.... .d. .
."
240 LET a$=a$+".m . ....m ..
. .... . . .hh.....o. .
."
250 LET a$=a$+"... m.....
..... . . . .hh..... p
z."
260 LET a$=a$+".....
.....
."
300 DIM m(20,5): DIM d(20,2)
310 FOR i=1 TO 20: FOR j=1 TO 5
: READ a: LET m(i,j)=a: NEXT j:
NEXT i
315 FOR i=1 TO 20: FOR j=1 TO 2
: READ a: LET d(i,j)=a: NEXT j:
NEXT i
320 DATA 8,3,7,5,2,7,5,8,5,3,7,
6,8,9,7,6,8,9,7,8,9,7,5,6,2,3,5,
1,6,8,9,4,3,5,4,6,7,3,5,8,2,3,9,
10,3,7,6,9,10,11,12,3,5,6,8,7,2,
4,3,5,9,8,9,7,6,7,9,5,10,12,3,2,
10,12,14,12,2,4,3,5,7,6,9,9,9,6,
5,6,1,5,3,2,4,5,7,9,24,35,43,56
330 DATA 3,4,5,3,7,8,6,9,10,2,5
,9,8,6,7,4,5,12,13,14,15,21,22,2
4,25,31,9,9,8,7,6,5,6,7,6,5,4,5,
6,7
900 GO SUB 9500
1000 PAPER 1: INK 6: BORDER 1: C
LS

```

```

1001 LET d$="SOUTH": LET cd1=0:
LET cd2=0: LET cd3=0: LET rd=0
1002 LET hj=0: LET mark=0: LET c
ontrol=0
1003 LET door1=0: LET door3=0
1005 PLOT 21,170: DRAW 213,0: DR
AW 0,-11: DRAW -213,0: DRAW 0,11
1010 PLOT 95,104: DRAW 65,0: DRA
W 0,-33: DRAW -65,0: DRAW 0,33
1020 PLOT 88,120: DRAW 79,0: PLO
T 88,55: DRAW 79,0: PLOT 175,143
: DRAW 0,-119: PLOT 173,145: DRA
W 0,-123: PLOT 80,143: DRAW 0,-1
19: PLOT 82,145: DRAW 0,-123
1030 INK 4: PLOT 5,90: DRAW 69,0
: DRAW 0,-65: DRAW -69,0: DRAW 0
,65
1040 INK 4: PLOT 7,143: DRAW 57,
0: PLOT 183,143: DRAW 57,0
1050 INK 7: BRIGHT 1: PRINT AT 1
7,23;"Shields";AT 18,28;"AoK";AT
 3,1;"Scanner";AT 15,23;"Comput
er";AT 16,28;"AoK";AT 6,1;"Statu
s";AT 7,6;"AoK";AT 9,1;"Report";
AT 11,1;"No";AT 12,1;"Contact";A
T 3,23;"Systems";AT 5,23;"Weapon
s";AT 6,28;"AoK";AT 7,23;"Drive"
;AT 8,28;"AoK";AT 9,23;"Visual";
AT 10,28;"AoK";AT 11,23;"Scanner
";AT 12,28;"AoK";AT 13,23;"Life
Sup";AT 14,28;"AoK"
1060 PRINT AT 16,14;"MODE";AT 18
,12;"COMPUTER"
1100 LET c$="Joystick or Keyboar
d?": GO SUB 3000:
1120 IF INKEY$="0" THEN LET mnb
=1: GO TO 1500
1130 IF INKEY$="k" THEN LET mnb
=0: GO TO 1500
1140 GO TO 1110
1500 LET d$="SOUTH": LET o=52.
1510 LET aok=100: LET ar=100: LE
T sok=100: LET wok=100: LET dok=
100: LET vok=100: LET lok=100: L
ET cok=100

```



```

1520 LET we=wok: LET dr=dok: LET
    vi=vok: LET sc=sok: LET li=lok:
    LET co=cok
1600 GO SUB 2000
1601 GO TO 4000
1605 REM
1610 REM check for an encounter
1615 REM
1620 IF a$(o)<>"d" THEN GO TO 1
700
1625 INK 7: BRIGHT 1: PRINT AT 1
1,1;"Ship      ";AT 12,1;"Defence
"

1630 LET c$="ALERT! ALERT! Ship
defence mechanism in operation":
GO SUB 3000
1635 LET ef=d(INT (20*RND)+1,1):
    LET ed=INT (RND*11)+1: LET es=9
: LET ea=d(INT (RND*20)+1,2)
1640 GO SUB 3500
1650 LET c$="We have destroyed t
he ship defence mechanism!": GO
SUB 3000
1660 PRINT AT 11,1;"No      ";AT
12,1;"Contact "
1670 LET c$="Returning to COMPUT
ER mode": GO SUB 3000
1680 PRINT AT 18,12;"COMPUTER"
1685 LET a$(o)=" "
1690 GO TO 4000
1700 IF a$(o)<>"g" THEN GO TO 1
750
1705 PRINT AT 11,1;"Guardian";AT
12,1;"'Droid  ": LET c$="Alert!
Alert! Guardian 'Droid on scann
er": GO SUB 3000
1710 LET es=11: LET ed=12: LET e
f=10: LET ea=8: GO SUB 3500
1720 LET c$="We destroyed the Gu
ardian 'Droid!": GO SUB 3000
1730 PRINT AT 11,1;"No      ";AT
12,1;"Contact "
1745 LET c$="Returning to COMPUT
ER mode": GO SUB 3000: PRINT AT

```

```

18,12;"COMPUTER";AT 11,1;"No
";AT 12,1;"Contact": LET a$(o)
=" ": GO TO 4000
1750 IF a$(o)<>"m" THEN GO TO 1
770
1751 PRINT AT 11,1;"Mute";AT 12,
1;"
1752 LET c$="Alert! Alert! Mute
on scanner          Blasti
ng him out of existence now": GO
SUB 3000: LET cho=(INT (RND*20)
+1): LET ea=m(cho,5): LET es=m(c
ho,3): LET ef=m(cho,1): LET ed=m
(cho,2): LET es=m(cho,4): GO SUB
3500
1755 LET c$="We destroyed the mu
te!": GO SUB 3000
1760 LET c$="Returning to COMPUT
ER mode": GO SUB 3000: PRINT AT
18,12;"COMPUTER";AT 11,1;"No
";AT 12,1;"Contact": LET a$(o)
=" ": GO TO 4000
1770 IF a$(o)<>"y" THEN GO TO 1
780
1772 PRINT AT 11,1;"Slimey";AT 1
2,1;"Mess          ": LET c$="Slimey m
ess on scanner": GO SUB 3000: LE
T sm=1: GO TO 4510
1780 IF a$(o)<>"f" THEN GO TO 1
790
1782 LET c$="We are in some radi
oactive fog          It is drainin
g power from the Drive and Shiel
d systems          Standby for dama
ge report": GO SUB 3000
1784 LET dy=dr-INT (RND*15)+1: L
ET c$="Drive reduced by ": LET c
$=c$+STR$ (100-dy): LET c$=c$+"%
": GO SUB 3000: LET dr=dy: IF dr
<10 THEN PRINT AT 8,28;"00%": L
ET e$="Drive": GO TO 9900
1786 PRINT AT 8,28;dr;"%": IF dr
>=100 THEN PRINT AT 8,28;"AoK"
1787 LET ay=ar-INT (RND*15)+1: L

```

```

ET c$="Shields reduced by ": LET
c$=c$+STR$ (100-ay): LET c$=c$+
"%": GO SUB 3000: LET ar=ay: IF
ar<10 THEN PRINT AT 8,28;"00%":
LET e$="Shields": GO TO 9900
1788 PRINT AT 18,28;ar;"%": IF a
r<>100 THEN PRINT AT 18,28;"AoK
"

1789 LET c$="Returning to COMPUT
ER mode": GO SUB 3000: PRINT AT
18,12;"COMPUTER";AT 11,1;"No
";AT 12,1;"Contact": LET a$(o)
=" ": GO TO 4000
1790 IF a$(o)<>"h" THEN GO TO 1
800
1792 LET c$="We are now on top o
f a sheet of liquid helium and a
re out of control We smash
ed into the side of a wall
Shields held but sustained dama
ge of ": LET ay=ar-INT (RND*15)+
1: LET c$=c$+STR$ (100-ay): LET
ar=ay: LET c$=c$+"%": GO SUB 300
0
1794 IF ar<10 THEN PRINT AT 18,
28;"00%": LET e$="Shields": GO T
O 9900
1796 PRINT AT 18,28;ar;"%": IF a
r>=100 THEN PRINT AT 18,28;"AoK
"

1798 GO TO 4510
1800 IF a$(o+hj)<>"o" THEN GO T
O 1810
1802 LET c$="In front of us is a
door into a restricted area
It can only be opened by CO
MMAND DISC 1": GO SUB 3000
1804 LET c$="Returning to COMPUT
ER mode": GO SUB 3000: PRINT AT
18,12;"COMPUTER":
1805 GO TO 4000
1810 IF a$(o)<>"1" THEN GO TO 1
820

```

```

1812 LET c$="Alert! Alert! Mute
on scanner He appears to be ca
rrying a command disc ": GO SU
B 3000: PRINT AT 11,1; INK 7; BR
IGHT 1;"Mute";AT 12,1;"
1813 LET cho=(INT (RND*20)+1): L
ET ea=m(cho,5): LET ef=m(cho,1):
LET ed=m(cho,2): LET es=m(cho,4
): GO SUB 3500
1814 PRINT AT 11,1;"No ";AT
12,1;"Contact"
1815 LET c$="We destroyed the mu
te! ": GO SUB 3000: LET a$(o)
="7": LET c$="On the ground ther
e is a command disc 1": GO SUB 3
000
1817 LET c$="Returning to COMPUT
ER mode ": GO SUB 3000: PRINT AT
18,12;"COMPUTER": GO TO 4000
1820 IF a$(o)<>"r" THEN GO TO 1
830
1822 LET c$="Alert! Alert! There
is a LETHAL looking robot on sc
anner He appears to be carryin
g a command disc ": GO SUB 300
0: PRINT AT 11,1; INK 7; BRIGHT
1;"Lethal";AT 12,1;"Looking";AT
13,1;"Robot"
1824 LET ea=9: LET es=7: LET ef=
10: LET ed=8
1825 GO SUB 3500
1827 LET c$="We destroyed the Ro
bot ": GO SUB 3000: LET c$="On
the ground there is a command di
sc 2": GO SUB 3000: PRINT AT 11,
1;"No ";AT 12,1;"Contact";AT
13,1;"
1828 LET a$(o)="8": LET c$="Retu
rning to COMPUTER mode ": GO SUB
3000: PRINT AT 18,12;"COMPUTER"
: GO TO 4000
1830 IF a$(o)<>"i" THEN GO TO 1
840
1832 PRINT AT 11,1;"Guardian";AT

```

```

12,1;"Droid ": LET c$="Alert!
Alert! Guardian 'Droid on scann
er He appears to be carryin
g a command disc ": GO SUB 3000
1834 LET es=10: LET e=7: LET ef=
8: LET ed=9: GO SUB 3500
1835 LET c$="We have destroyed t
he Guardian Droid On the
ground there is a command disc 3
": GO SUB 3000: LET a$(o)="9":
PRINT AT 11,1;"No ";AT 12,1
;"Contact"
1837 LET c$="Returning to COMPUT
ER mode ": GO SUB 3000: PRINT AT
18,12;"COMPUTER": GO TO 4000
1840 IF a$(o+hj)<>"p" THEN GO T
O 1850
1842 LET c$="In front of us is a
door into a restricted area
It can only be opened by CO
MMAND DISC 3": GO SUB 3000
1843 LET c$="Returning to COMPUT
ER mode ": GO SUB 3000: PRINT AT
18,12;"COMPUTER": GO TO 4000
1844 GO TO 4000
1850 IF a$(o)<>"n" THEN GO TO 1
860
1852 LET c$="There appears to be
a friendly mutant sitting in fr
ont of us He gives us a Robot
Disrupter which can be used to
destroy all robots in the repair
shop area of the ship with
out this we would not come out a
live He also tells us tha
t we are in the crews quarters w
ith only one way out Turn
around and take the first on the
right With that he dsiappe
ars ": GO SUB 3000: LET r
d=1: LET a$(o)=" "
1854 GO TO 4500
1860 IF a$(o)<>"w" THEN GO TO 1
870

```

```
1862 LET c$="We appear to have come across a crude form of Radio active waste The engines and weapons systems are using it to recharge": GO SUB 3000
1864 LET we=100: LET dr=100
1866 PRINT AT 6,28;"AoK";AT 8,28;"AoK"
1867 LET c$="Returning to DRIVE mode": GO SUB 3000: GO TO 4510
1868 GO TO 4000
1870 IF a$(o)<>"a" THEN GO TO 1880
1872 LET c$="We appear to have stumbled upon 5 mutants having a card game A bad move Our only chance is running If the drive is powerful enough": GO SUB 3000
1874 IF dr>50 THEN LET c$="It is though we have moved into a dangerous place Possibly more dangerous": GO SUB 3000: LET o=1
185: LET c$="Returning to COMPUTER mode ": GO SUB 3000: PRINT AT 18,12;"COMPUTER": GO TO 4000
1876 LET c$="Not fast enough We have been caught": GO TO 3000: LET li=0: LET e$="Life Support": PRINT AT 14,28;"00%": GO TO 9900
1880 IF a$(o)<>"b" THEN GO TO 1890
1882 LET c$="We appear to have entered some sort of robot repair plant A very bad move All the thousands of robots have been activated and are heading our way ": GO SUB 3000
1884 IF rd=1 THEN LET c$="The Robot Disrupter has suddenly activated all by itself It is dealing with the Robots so that we can make our getaway ": GO SUB 3000: LET o=286: LET d$="EAST": GO SUB 2000: GO TO 4510
```

```

1885 LET c$="We appear to have a
      problem": GO SUB 3000
1886 LET es=100: LET ea=40: LET
ed=90: LET ef=100: GO SUB 3500
1890 IF a$(o)<>"c" THEN GO TO 1
900
1892 INK 7: BRIGHT 1: PRINT AT 1
1,1;"Nuclear";AT 12,1;"Reactor":
      LET c$="Alert! Alert! Nuclear r
eactor on scanner  It is draini
ng power from the life support s
ystem ": GO SUB 3000
1893 LET li=li-INT (RND*20): PRI
NT AT 14,28;li;"%": IF li>=100 T
HEN PRINT AT 14,28;"AoK "
1894 IF li<10 THEN LET li=0: PR
INT AT 14,27;" 00%": LET e$="Lif
e Support": GO TO 9900
1896 IF RND>.5 THEN LET c$="It
has taken too much power and exp
loaded      Shields holding but t
hey have sustained damage": GO S
UB 3000: LET ar=ar-INT (RND*15)
1897 IF ar<10 THEN LET ar=0: LE
T e$="Shields": PRINT AT 14,27;"
 00%": GO TO 9900
1898 PRINT AT 14,28;ar;"%": IF a
r>=100 THEN PRINT AT 14,28;"AoK
"
1899 GO TO 4510
1900 IF a$(o)<>"d" THEN GO TO 1
910
1902 PRINT AT 11,1; INK 7; BRIGH
T 1;"Sentry";AT 12,1;"'Droid "
1904 LET es=10: LET ea=7: LET ef
=6: LET ed=8
1906 LET c$="We destroyed the Se
ntry 'Droid  Returning to DRIVE
mode      ": GO SUB 3000: PRINT AT
18,12;" DRIVES "
1908 GO TO 4510
1910 IF a$(o)<>"t" THEN GO TO 1
940

```

```

1911 INK 1: FOR i=9 TO 12: PRINT
    AT i,12;"          ": NEXT i
1912 INK 7: BRIGHT 1: PRINT AT 1
1,1;"TELE-";AT 12,1;"    PORT"
1914 INK 4: BRIGHT 1: PLOT 100,1
02: DRAW 56,0: DRAW 0,-29: DRAW
-18,0: DRAW -10,14: DRAW -10,-14
: DRAW -18,0: DRAW 0,29
1916 PLOT 123,93: DRAW 10,0: DRA
W 0,-6: DRAW -10,0: DRAW 0,6
1918 PLOT 102,100: DRAW 52,0: DR
AW 0,-24: DRAW -14,0: DRAW -12,1
5: DRAW -12,-15: DRAW -13,0: DRA
W 0,24
1919 PLOT 108,97: DRAW 4,0: DRAW
0,-2: DRAW -4,0: DRAW 0,2
1920 PLOT 108,93: DRAW 4,0: DRAW
0,-2: DRAW -4,0: DRAW 0,2
1921 PLOT 108,89: DRAW 4,0: DRAW
0,-2: DRAW -4,0: DRAW 0,2
1922 PLOT 108,85: DRAW 4,0: DRAW
0,-2: DRAW -4,0: DRAW 0,2
1923 PLOT 108,81: DRAW 4,0: DRAW
0,-2: DRAW -4,0: DRAW 0,2
1924 PLOT 144,96: DRAW 0,-15: DR
AW 3,0: DRAW 0,15: DRAW -1,0: DR
AW 0,-14: DRAW -1,0: DRAW 0,14
1926 LET c$="In front of us stan
ds a TELEPORT control panel
A slot is ready to accept a comm
and disc": GO SUB 3000
1928 LET tele=1: GO TO 4000
1930 IF q$(4)(1)<>"2" THEN LET
c$="That Command Disc does not o
perate the teleport    It is ret
urned          ": GO SUB 3000: GO T
O 4000
1932÷LET c$="The control panel h
ums into life          The disc
is returned and we are transpor
ted into the command part of the
ship":GO SUB 3000
1934 INK 7: BRIGHT 1: PRINT AT 1
1,1;"No          ";AT 12,1;"Contact":

```



```

LET o=477: GO SUB 2000: GO TO 4
000
1940 IF a$(o)<>"x" THEN GO TO 9
100
1942 INK 1: FOR i=9 TO 12: PRINT
AT i,12;"          ": NEXT i
1944 INK 7: BRIGHT 1: PRINT AT 1
1,1;"CONTROL ";AT 12,1;"PANEL "
1946 INK 4: BRIGHT 1: PLOT 100,1
02: DRAW 56,0: DRAW 0,-29: DRAW
-56,0: DRAW 0,29
1948 PLOT 102,100: DRAW 52,0: DR
AW 0,-24: DRAW -52,0: DRAW 0,24
1950 PLOT 110,96: DRAW 0,-15: DR
AW 3,0: DRAW 0,15: DRAW -1,0: DR
AW 0,-14: DRAW -1,0: DRAW 0,14
1952 PLOT 126,96: DRAW 0,-15: DR
AW 3,0: DRAW 0,15: DRAW -1,0: DR
AW 0,-14: DRAW -1,0: DRAW 0,14
1954 PLOT 145,96: DRAW 0,-15: DR
AW 3,0: DRAW 0,15: DRAW -1,0: DR
AW 0,-14: DRAW -1,0: DRAW 0,14
1960 LET c$="In front of us ther
e is a control panel with three
slots in it": GO SUB 3000
1962 GO TO 4000
1964 PRINT AT 10,13; INK 7; BRIG
HT 1; FLASH 1;"ACTIVE"; FLASH 0:
LET c$="Mark Activated heading
for escape pod": GO SUB 3000
1966 LET d$="EAST": LET o=1046:
GO SUB 2000: LET o=1047: GO SUB
2000: LET o=1048: GO SUB 2000: L
ET o=1049: GO SUB 2000: LET d$="
SOUTH": LET o=1099: GO SUB 2000:
LET o=1149: GO SUB 2000: LET o=
1199: GO SUB 2000: GO TO 9100
2000 INK 1: FOR i=9 TO 12: PRINT
AT i,12;"          ": NEXT i
2005 INK 7: BRIGHT 0
2010 IF d$<>"SOUTH" THEN GO TO
2050
2023 PLOT 95,104: DRAW 20,-10: D
RAW 0,-12: DRAW -20,-11: IF a$(o

```

```

+1)<>". " THEN PLOT 100,101: DRA
W 0,-28: PLOT 100,96: DRAW 9,0:
DRAW 0,-18: DRAW -9,0
2025 PLOT 160,104: DRAW -20,-10:
DRAW 0,-12: DRAW 20,-11: IF a$(
o-1)<>". " THEN PLOT 155,101: DR
AW 0,-27: PLOT 155,97: DRAW -9,0
: DRAW 0,-18: DRAW 9,0
2030 IF a$(o+50)="." THEN PLOT
115,94: DRAW 25,0: PLOT 115,82:
DRAW 25,0
2040 RETURN
2050 IF d$<>"NORTH" THEN GO TO
2100
2055 PLOT 95,104: DRAW 20,-10: D
RAW 0,-12: DRAW -20,-11: IF a$(o
-1)<>". " THEN PLOT 100,101: DRA
W 0,-28: PLOT 100,96: DRAW 9,0:
DRAW 0,-18: DRAW -9,0
2060 PLOT 160,104: DRAW -20,-10:
DRAW 0,-12: DRAW 20,-11: IF a$(
o+1)<>". " THEN PLOT 155,101: DR
AW 0,-27: PLOT 155,97: DRAW -9,0
: DRAW 0,-18: DRAW 9,0
2070 IF a$(o-50)="." THEN PLOT
115,94: DRAW 25,0: PLOT 115,82:
DRAW 25,0
2080 RETURN
2100 IF d$<>"EAST" THEN GO TO 2
150
2110 PLOT 95,104: DRAW 20,-10: D
RAW 0,-12: DRAW -20,-11: IF a$(o
-50)<>". " THEN PLOT 100,101: DR
AW 0,-28: PLOT 100,96: DRAW 9,0:
DRAW 0,-18: DRAW -9,0
2120 PLOT 160,104: DRAW -20,-10:
DRAW 0,-12: DRAW 20,-11: IF a$(
o+50)<>". " THEN PLOT 155,101: D
RAW 0,-27: PLOT 155,97: DRAW -9,
0: DRAW 0,-18: DRAW 9,0
2130 IF a$(o+1)="." THEN PLOT 1
15,94: DRAW 25,0: PLOT 115,82: D
RAW 25,0
2140 RETURN

```

```

2150 PLOT 95,104: DRAW 20,-10: D
RAW 0,-12: DRAW -20,-11: IF a$(o
+50)<>"." THEN PLOT 100,101: DR
AW 0,-28: PLOT 100,96: DRAW 9,0:
DRAW 0,-18: DRAW -9,0
2160 PLOT 160,104: DRAW -20,-10:
DRAW 0,-12: DRAW 20,-11: IF a$(
o-50)<>"." THEN PLOT 155,101: D
RAW 0,-27: PLOT 155,97: DRAW -9,
0: DRAW 0,-18: DRAW 9,0
2170 IF a$(o-1)="." THEN PLOT 1
15,94: DRAW 25,0: PLOT 115,82: D
RAW 25,0
2180 RETURN
2190 STOP
2922 LET c$=c$+STR$ b: LET c$=c$
+"%": GO SUB 3000
3000 INK 7: BRIGHT 1: FOR i=1 TO
LEN c$
3010 IF i>25 THEN GO TO 3100
3020 PRINT AT 1,29-i;c$(1 TO i)
3025 BEEP 0.005,-12
3030 NEXT i
3040 GO TO 3200
3100 FOR m=1 TO LEN c$-25
3120 PRINT AT 1,3;c$(m TO m+25)
3125 BEEP 0.005,-12
3130 NEXT m
3200 FOR i=1 TO 100: NEXT i: PRI
NT AT 1,3;"
"
3210 RETURN
3500 LET c$="Weapons systems act
ivated": GO SUB 3000: INK 7: BRI
GHT 1: PRINT AT 18,12;" WEAPONS"
3505 LET os=INT (li/10)
3510 LET od=INT (dr/10)
3520 LET of=INT (we/10)
3530 LET oa=INT (ar/10)
3540 LET ias=os+of
3550 LET ieas=es+ef
3555 LET as=ias+INT (RND*12)+1
3556 LET eas=ieas+INT (RND*12)+1
3560 IF INT (RND*10)<8 THEN GO
TO 3580

```

```

3565 LET as=as+2: LET c$="We surprised it!": GO SUB 3000
3570 GO TO 3600
3580 IF INT (RND*10)>2 THEN GO TO 3600
3585 LET eas=eas+2: LET c$="It surprised us!": GO SUB 3000
3620 IF as<eas THEN LET c$="It": GO TO 3635
3630 LET c$="We"
3635 LET rnd=RND*10
3640 IF rnd>9 THEN LET c$=c$+"fired a rapid burst of TYLIRIM bolts": GO SUB 3000: GO TO 3750
3650 IF rnd>8 THEN LET c$=c$+"fired a searing laser blast": GO SUB 3000: GO TO 3750
3660 IF rnd>7 THEN LET c$=c$+"saturated the room with laser fire": GO SUB 3000: GO TO 3750
3670 IF rnd>6 THEN LET c$=c$+"blasted out a beam of DEUTIRIUM": GO SUB 3000: GO TO 3750
3680 IF rnd>5 THEN LET c$=c$+"carved a hole in the armour with a sufficiently powerful physmic bolt": GO SUB 3000: GO TO 3750
3690 IF rnd>4 THEN LET c$=c$+"sent a shock of devastating power": GO SUB 3000: GO TO 3750
3700 IF rnd>3 THEN LET c$=c$+"flung a massive radiation bolt": GO SUB 3000: GO TO 3750
3710 IF rnd>2 THEN LET c$=c$+"hurled a shuddering COLLAPSIUM bolt": GO SUB 3000: GO TO 3750
3720 IF rnd>1 THEN LET c$=c$+"fired a sharp beam of IGNITRON": GO SUB 3000: GO TO 3750
3730 LET c$=c$+"fired a pulsating ray of TYLIRIUM": GO SUB 3000
3750 IF as<eas THEN GO TO 3810
3760 IF RND*20>ea THEN LET es=es-2

```

```

3765 IF as-eas>4 THEN LET ea=ea
-2
3766 IF as-eas>6 THEN LET ef=ef
-INT (RND*5)
3767 LET ed=ed-INT (RND*5)
3768 IF es<=0 OR ef<=0 OR ed<=0
OR ea<=0 THEN RETURN
3770 LET c$=" We have damaged it
!": GO SUB 3000
3780 GO TO 3955
3810 IF RND>.6 THEN LET c$="No
damage incurred Shields holding"
: GO SUB 3000: GO TO 3505
3815 LET c$="It has damaged us.
DAMAGE REPORT follows....": GO S
UB 3000
3820 LET ly=li-INT (RND*20)+1: L
ET c$="Life support damaged by "
: LET b=ABS (100-INT ((li/ly*100
)))
3821 IF b>100 THEN LET b=100
3822 LET c$=c$+STR$ b: LET c$=c$
+"%": GO SUB 3000
3823 IF ly<10 THEN LET e$="Life
Support": PRINT AT 14,28;"00%":
GO TO 9900
3825 LET li=ly: PRINT AT 14,28;I
NT li;"%": IF li>=100 THEN PRIN
T AT 14,28;"AoK "
3830 LET dy=dr-INT (RND*10)+1: L
ET c$="Drive damaged by ": LET b
=ABS (100-INT ((dr/dy*100)))
3831 IF b>100 THEN LET b=100
3832 LET c$=c$+STR$ b: LET c$=c$
+"%": GO SUB 3000
3834 IF dy<10 THEN LET e$="Driv
e": PRINT AT 8,28;"00%": GO TO 9
900
3835 LET dr=dy: PRINT AT 8,28;IN
T dr;"%": IF dr>=100 THEN PRINT
AT 8,28;"AoK "
3840 LET wy=we-INT (RND*20)+1: L
ET c$="Weapons systems damaged b
y ": LET b=ABS (100-INT ((wy/we*
100)))

```

```

3842 IF b>100 THEN LET b=100
3843 LET c$=c$+STR$ b: LET c$=c$
+"%": GO SUB 3000
3845 IF wy<10 THEN LET we=wy: P
RINT AT 6,28;"00%": LET e$="WEAP
ONS": GO TO 9900
3850 LET we=wy: PRINT AT 6,28;we
;"%": IF we>=100 THEN PRINT AT
6,28;"AoK "
3880 LET vy=vi-INT (RND*20)+1: L
ET c$="Visual systems damaged by
": LET b=ABS (100-INT ((vy/vi*1
00)))
3881 IF b>100 THEN LET b=100
3882 LET c$=c$+STR$ b: LET c$=c$
+"%": GO SUB 3000
3885 IF vy<10 THEN LET e$="Visu
al": PRINT AT 10,28;"00%": GO TO
9900
3890 LET vi=vy: PRINT AT 10,28;v
i;"%": IF vi>=100 THEN PRINT AT
10,28;"AoK "
3900 LET sy=sc-INT (RND*20)+1: L
ET c$="Scanner systems damaged b
y ": LET b=ABS (100-INT ((sy/sc*
100)))
3901 IF b>100 THEN LET b=100
3902 LET c$=c$+STR$ b: LET c$=c$
+"%": GO SUB 3000
3905 IF sy<10 THEN LET e$="Scan
ner": PRINT AT 12,28;"00%";AT 7,
6;"00%": GO TO 9900
3910 LET sc=sy: PRINT AT 12,28;s
c;"%";AT 7,6;sc;"%": IF sc>=100
THEN PRINT AT 12,28;"AoK ";AT 7
,6;"AoK "
3920 LET cy=co-INT (RND*20)+1: L
ET c$="Computer damaged by ": L
ET b=ABS (100-INT ((cy/co*100)))
3921 IF b>100 THEN LET b=100
3922 LET c$=c$+STR$ b: LET c$=c$
+"%": GO SUB 3000
3925 IF cy<10 THEN LET e$="Comp
uter": PRINT AT 16,28;"00%": GO
TO 9900

```

```

3930 LET co=cy: PRINT AT 16,28;c
o;"%": IF co>=100 THEN PRINT AT
16,28;"AoK "
3940 LET ay=ar-INT (RND*20)+1: L
ET c$="Shield pattern disrupted
by ": LET b=ABS (100-INT ((ay/ar
*100)))
3941 IF b>100 THEN LET b=100
3942 LET c$=c$+STR$ b: LET c$=c$
+"%": GO SUB 3000
3945 IF ay<10 THEN LET e$="Shie
lds": PRINT AT 18,28;"00%": GO T
O 9900
3950 LET ar=ay: PRINT AT 18,28;a
r;"%": IF ar>=100 THEN PRINT AT
18,28;"AoK "
3955 IF es=0 OR ef=0 OR ea=0 OR
ed=0 THEN RETURN
3960 GO TO 3505
4000 DIM q$(12,15)
4005 LET i$=""
4010 PRINT AT 1,28;"-"
4020 IF mnb=1 AND INKEY$="O" THE
N GO TO 4500
4021 LET k$=INKEY$: IF k$="" THE
N GO TO 4021
4025 IF CODE k$=13 THEN PRINT A
T 1,28;" ": LET i$=i$+" ": GO TO
4050
4026 IF CODE k$=12 AND LEN i$÷0
THEN LET i$=i$(1 TO LEN i$-1):
GO TO 4032
4027 IF CODE k$=12 THEN GO TO 4
032
4030 LET i$=i$+k$
4031 IF LEN i$>=95 THEN LET c$=
"Sorry, command too long, retype
please": GO SUB 3000: GO TO 400
0
4032 PRINT AT 1,3;"
"
4035 IF LEN i$>24 THEN PRINT AT
1,3;i$(LEN i$-24 TO LEN i$);"-
": GO TO 4020

```

```

4040 PRINT AT 1,28-LEN i$ ;i$;"-
": GO TO 4020
4050 LET last=1: LET v=1: FOR i=
1 TO LEN i$
4060 IF i$(i)=" " THEN LET q$(v
)=i$(last TO i-1): LET last=i+1:
LET v=v+1
4070 NEXT i
4080 PRINT AT 1,3;"
"

4095 REM
4100 REM definate article elimin
ator
4105 REM
4110 FOR h=1 TO v-1
4120 IF q$(h)(1 TO 3)="the" THEN
FOR i=h+1 TO v: LET q$(i-1)=q$(
i): NEXT i
4130 NEXT h
4200 REM check for keywords
4210 IF q$(1)(1 TO 4)="open" THE
N GO TO 4300
4220 IF q$(1)(1 TO 6)="change" T
HEN GO TO 4500
4250 IF q$(1)(1 TO 9)="inventory
" THEN GO TO 4600
4260 IF q$(1)(1 TO 4)="wait" THE
N GO TO 4700
4265 IF q$(1)(1 TO 5)="pause" TH
EN GO TO 4700
4270 IF q$(1)(1 TO 4)="save" THE
N GO TO 4900
4275 IF q$(1)(1 TO 4)="load" THE
N GO TO 5000
4280 IF q$(1)(1 TO 4)="take" THE
N GO TO 5100
4285 IF q$(1)(1 TO 6)="insert" T
HEN GO TO 5200
4286 IF q$(1)(1 TO 4)="mark" THE
N GO TO 5300
4290 LET c$="I do not understand
could you please retype": GO SU
B 3000: GO TO 4000
4300 IF a$(o+hj)<>"o" AND a$(o+h

```



```
j)<>"p" THEN LET c$="I see nothing to open": GO SUB 3000: GO TO 4000
4310 IF q$(3)(1 TO 4)<>"with" THEN GO TO 4290
4320 IF q$(4)(1 TO 7)="command" AND q$(5)(1 TO 4)="disc" THEN GO TO 4340
4330 LET c$="You cannot open a door with that": GO SUB 3000: GO TO 4000
4340 IF q$(6)(1 TO 1)="1" AND cd1=1 AND a$(o+hj)="o" THEN LET c$="The door slides swiftly to one side": GO SUB 3000: LET a$(o+hj)=" ": LET a$(o+hj+hj)=" ": GO TO 4000
4350 IF q$(6)(1 TO 1)="1" AND cd1=0 THEN LET c$="We do not possess command disc 1": GO SUB 3000: GO TO 4000
4360 IF q$(6)(1 TO 1)="2" THEN LET c$="Command disc 2 does not control the door opening system": GO SUB 3000: GO TO 4000
4365 IF q$(6)(1 TO 1)="3" AND cd3=1 AND a$(o+hj)="p" THEN LET c$="The door slides swiftly to one side": GO SUB 3000: LET a$(o+hj)=" ": LET a$(o+hj+hj)=" ": GO TO 4000
4370 IF q$(6)(1 TO 1)="3" AND cd3=0 THEN LET c$="We do not possess command disc 3": GO SUB 3000: GO TO 4000
4380 IF q$(6)(1 TO 1)="1" AND a$(o+hj)="p" THEN LET c$="You cannot open this door with command disc 1": GO SUB 3000: GO TO 4000
4390 IF q$(6)(1 TO 1)="3" AND a$(o+hj)="o" THEN LET c$="You cannot open this door with command disc 3": GO SUB 3000: GO TO 4000
4395 LET c$="I do not understand
```

```

please retype": GO SUB 3000: GO
TO 4000
4500 LET c$="Changing into DRIVE
mode ":GO SUB 3000:PRINT A
T 18,12;INK 7;BRIGHT 1;" DRIVES
":LET c$="Drive on line Dr
ive systems at ":LET c$=c$+STR$
dr:LET c$=c$+"% ":GO SUB 3000
4510 IF d$="SOUTH" THEN GO TO 7
100
4520 IF d$="NORTH" THEN GO TO 7
200
4530 IF d$="EAST" THEN GO TO 73
00
4540 IF d$="WEST" THEN GO TO 74
00
4600 LET c$="At this moment we h
ave in the hold :- ": IF rd=1 TH
EN LET c$=c$+"A Robot Disrupter
"
4605 IF cd1=1 THEN LET c$=c$+"C
ommand disc 1 "
4610 IF cd2=1 THEN LET c$=c$+"C
ommand disc 2 "
4615 IF cd3=1 THEN LET c$=c$+"C
ommand disc 3 "
4620 IF rd=0 AND cd1=0 AND cd2=0
AND cd3=0 THEN LET c$=c$+"noth
ing"
4630 GO SUB 3000
4640 GO TO 4000
4700 LET c$="Shutting down all s
ystems now To reactivate hi
t any key": GO SUB 3000
4710 INK 4: FOR i=103 TO 72 STEP
-1: PLOT 96,i: DRAW 63,0: NEXT
i
4715 PRINT AT 18,12; INK 7; BRIG
HT 1;"SLEEPING"
4720 IF INKEY$="" THEN GO TO 47
20
4730 INK 1: FOR i=9 TO 12: PRINT
AT i,12;" ": NEXT i
4740 GO SUB 2000

```

```

4750 LET c$="COMPUTER on line  "
      : GO SUB 3000
4760 PRINT AT 18,12;"COMPUTER"
4770 GO TO 4000
4900 DIM w(14)
4905 LET w(14)=mnb
4910 LET w(1)=cd1: LET w(2)=cd2:
      LET w(3)=cd3: LET w(4)=rd
4920 LET w(5)=o: LET w(6)=dr: LE
T w(7)=ar: LET w(8)=co: LET w(9)
=we: LET w(10)=vi: LET w(11)=li:
      LET w(12)=sc
4930 SAVE "Mute Data" DATA w()
4940 LET c$="Do you wish to cont
inue (y/n)": GO SUB 3000
4945 IF INKEY$="y" THEN GO TO 4
000
4950 IF INKEY$="n" THEN FOR i=1
03 TO 72 STEP -1: INK 4: BRIGHT
0: PLOT 96,i: DRAW 63,0: NEXT i:
      NEW
4955 GO TO 4945
5000 LET c$="Enter data cassette
and hit any key": GO SUB 3000
5010 LOAD "" DATA w()
5020 LET cd1=w(1): LET cd2=w(2):
      LET cd3=w(3): LET rd=w(4)
5030 LET o=w(5): LET dr=w(6): LE
T ar=w(7): LET co=w(8): LET we=w
(9): LET vi=(10): LET li=w(11):
LET sc=w(12)
5035 LET mnb=w(14)
5040 GO SUB 2000: GO TO 4000
5100 IF q$(2)(1 TO 7)<>"command"
OR q$(3)(1 TO 4)<>"disc" THEN
LET c$="I do not understand plea
se retype": GO SUB 3000: GO TO 4
000
5110 IF q$(4)(1)="1" AND a$(o)="
7" THEN LET a$(o)=" ": LET c$="
Command disc 1 has been picked u
p": GO SUB 3000: LET cd1=1: GO T
O 4000
5120 IF q$(4)(1)="2" AND a$(o)="

```

```

8" THEN LET a$(o)=" ": LET c$="
Command disc 2 has been picked u
p": GO SUB 3000: LET cd2=1: GO T
O 4000
5130 IF q$(4)(1)="3" AND a$(o)="
9" THEN LET a$(o)=" ": LET c$="
Command disc 3 has been picked u
p": GO SUB 3000: LET cd3=1: GO T
O 4000
5140 LET c$="That is not visible
at this time": GO SUB 3000: GO
TO 4000
5200 IF q$(4)(1)="2" AND cd2=1 A
ND a$(o)="t" THEN GO TO 1932
5210 IF a$(o)<>"x" THEN LET c$="
You cannot do that here": GO SU
B 3000: GO TO 4000
5215 IF mark=0 AND control=2 THE
N LET c$="Before you enter the
last disc you must find the esca
pe pod and mark it": GO SUB 3000
: GO TO 4000
5220 IF q$(4)(1)="1" THEN LET c
$="Command disc 1 is inserted":
GO SUB 3000: LET cd1=0: LET cont
rol=control+1: IF control=3 THEN
GO TO 1964
5225 IF q$(4)(1)="2" THEN LET c
$="Command disc 2 is inserted":
GO SUB 3000: LET cd2=0: LET cont
rol=control+1: IF control=3 THEN
GO TO 1964
5227 IF q$(4)(1)="3" THEN LET c
$="Command disc 3 is inserted":
GO SUB 3000: LET cd3=0: LET cont
rol=control+1: IF control=3 THEN
GO TO 1964
5230 GO TO 4000
5300 IF a$(o)="z" THEN LET mark
=1: LET c$="Escape pod marked N
ow go blow up the ship!": GO SUB
3000: GO TO 4000
5310 LET c$="No escape pod at th
is location": GO SUB 3000: GO TO
4000

```

```

7100 LET z$=INKEY$: IF z$="" THE
N   GO TO 7100
7101 IF (mnb=1 AND z$="0") OR z$
="r" THEN LET c$="Returning to
COMPUTER mode   ": GO SUB 3000
: PRINT AT 18,12;"COMPUTER": LET
c$="COMPUTER on line   ": GO SU
B 3000: GO TO 4000
7102 IF mnb=1 AND z$="7" THEN L
ET z$="8"
7103 IF mnb=1 AND z$="6" THEN L
ET z$="5"
7104 IF mnb=1 AND z$="8" THEN L
ET z$="6"
7105 IF mnb=1 AND z$="9" THEN L
ET z$="7"
7108 IF z$="5" OR z$="6" OR z$="
7" OR z$="8" THEN GO TO 7110
7109 GO TO 7100
7110 LET q=VAL z$
7115 IF q=7 AND a$(o+50)="." THE
N   GO TO 7100
7116 IF a$(o+50)="o" OR a$(o+50)
="p" THEN LET hj+=50: GO TO 161
0
7117 IF q=7 THEN LET o=o+50: GO
SUB 9990: GO SUB 2000: IF a$(o)
<>" " THEN GO TO 1610
7120 IF q=8 THEN LET d$="WEST":
GO SUB 2000
7130 IF q=5 THEN LET d$="EAST":
GO SUB 2000
7140 IF q=6 THEN LET d$="NORTH"
: GO SUB 2000
7150 GO TO 4510
7200 LET z$=INKEY$: IF z$="" THE
N   GO TO 7200
7201 IF (mnb=1 AND z$="0") OR z$
="r" THEN LET c$="Returning to
COMPUTER mode   ": GO SUB 3000
: PRINT AT 18,12;"COMPUTER": LET
c$="COMPUTER on line   ": GO SU
B 3000: GO TO 4000
7202 IF mnb=1 AND z$="6" THEN L
ET z$="5"

```

```

7203 IF mnb=1 AND z$="7" THEN L
ET z$="8"
7204 IF mnb=1 AND z$="8" THEN L
ET z$="6"
7205 IF mnb=1 AND z$="9" THEN L
ET z$="7"
7208 IF z$="5" OR z$="6" OR z$="
7" OR z$="8" THEN GO TO 7210
7209 GO TO 7200
7210 LET q=VAL z$
7215 IF q=7 AND a$(o-50)="." THE
N GO TO 7200
7216 IF a$(o-50)="o" OR a$(o-50)
="p" THEN : LET hj=-50: GO TO 16
10
7217 IF q=7 THEN LET o=o-50: GO
SUB 9990: GO SUB 2000: IF a$(o)
<>" " THEN GO TO 1610
7220 IF q=8 THEN LET d$="EAST":
GO SUB 2000
7230 IF q=5 THEN LET d$="WEST":
GO SUB 2000
7240 IF q=6 THEN LET d$="SOUTH"
: GO SUB 2000
7250 GO TO 4510
7300 LET z$=INKEY$: IF z$="" THE
N GO TO 7300
7301 IF (mnb=1 AND z$="0") OR z$
="r" THEN LET c$="Returning to
COMPUTER mode ": GO SUB 3000
: PRINT AT 18,12;"COMPUTER": LET
c$="COMPUTER on line ": GO SU
B 3000: GO TO 4000
7302 IF mnb=1 AND z$="6" THEN L
ET z$="5"
7303 IF mnb=1 AND z$="7" THEN L
ET z$="8"
7304 IF mnb=1 AND z$="8" THEN L
ET z$="6"
7305 IF mnb=1 AND z$="9" THEN L
ET z$="7"
7308 IF z$="5" OR z$="6" OR z$="
7" OR z$="8" THEN GO TO 7310
7309 GO TO 7300

```

```

7310 LET q=VAL z$
7315 IF q=7 AND a$(o+1)="." THEN
    GO TO 7300
7316 IF a$(o+1)="o" OR a$(o+1)="
p" THEN LET hj+=1: GO TO 1610
7325 IF q=7 THEN LET o=o+1: GO
SUB 9990: GO SUB 2000: IF a$(o)<
>" " THEN GO TO 1610
7330 IF q=8 THEN LET d$="SOUTH"
: GO SUB 2000
7340 IF q=5 THEN LET d$="NORTH"
: GO SUB 2000
7350 IF q=6 THEN LET d$="WEST":
GO SUB 2000
7360 GO TO 4510
7400 LET z$=INKEY$: IF z$="" THE
N GO TO 7400
7401 IF (mnb=1 AND z$="0") OR z$
="r" THEN LET c$="Returning to
COMPUTER mode ": GO SUB 3000
: PRINT AT 18,12;"COMPUTER": LET
c$="COMPUTER on line ": GO SU
B 3000: GO TO 4000
7402 IF mnb=1 AND z$="6" THEN L
ET z$="5"
7403 IF mnb=1 AND z$="7" THEN L
ET z$="8"
7404 IF mnb=1 AND z$="8" THEN L
ET z$="6"
7405 IF mnb=1 AND z$="9" THEN L
ET z$="7"
7408 IF z$="5" OR z$="6" OR z$="
7" OR z$="8" THEN GO TO 7410
7409 GO TO 7400
7410 LET q=VAL z$
7415 IF q=7 AND a$(o-1)="." THEN
GO TO 7400
7416 IF a$(o-1)="o" OR a$(o-1)="
p" THEN LET hj=-1: GO TO 1610
7417 IF q=7 THEN LET o=o-1: GO
SUB 9990: GO SUB 2000: IF a$(o)<
>" " THEN GO TO 1610
7420 IF q=8 THEN LET d$="NORTH"
: GO SUB 2000

```

```

7430 IF q=5 THEN LET d$="SOUTH"
: GO SUB 2000
7440 IF q=6 THEN LET d$="EAST":
GO SUB 2000
7450 GO TO 4510
9100 INK 1: FOR i=9 TO 12: PRINT
AT i,12;" ": NEXT i
9110 INK 7: BRIGHT 1: PRINT AT 1
1,1;"CONTROL ";AT 12,1;"PANEL
"
9120 INK 4: BRIGHT 1: PLOT 100,1
02: DRAW 56,0: DRAW 0,-29: DRAW
-56,0: DRAW 0,29
9130 CIRCLE 128,88,10
9140 CIRCLE 128,88,12
9145 CIRCLE 128,88,8
9146 CIRCLE 128,88,6
9147 CIRCLE 128,88,4
9150 LET c$="In front of us stan
ds a ESCAPE POD ": GO SUB 3000
9160 IF control<>3 THEN GO TO 4
000
9170 LET c$="The door opens and
we go in ": GO SUB 3000
9180 INK 1: FOR i=9 TO 12: PRINT
AT i,12;" ": NEXT i
9190 INK 5: PLOT 95,84: DRAW 65,
0: PLOT 122,84: DRAW -27,20: PLO
T 128,84: DRAW 32,20
9200 PLOT 125,94: PLOT 100,103:
PLOT 140,99: PLOT 115,90: PLOT 1
23,100: PLOT 146,97
9210 PLOT 105,84: DRAW 0,-5: DRA
W 45,0: DRAW 0,5: PLOT 110,84: D
RAW 0,-5: PLOT 145,84: DRAW 0,-5
9220 LET c$="EJECTING ": GO
SUB 3000
9230 FOR i=1 TO 40: BEEP 1/i,i:
NEXT i
9240 LET c$="Message just reciev
ed from starft/satcom(MUTE):- '
WELL DONE! A CRUISER IS BEING SE
NT OUT TO PICK YOU UP SEE Y
OU IN ABOUT THREE DAYS ": GO SUB
3000

```



```
9250 INK 1: FOR i=9 TO 12: PRINT
    AT i,12;"          ": NEXT i: PLO
T 95,104: DRAW 65,0: DRAW 0,-33:
    DRAW -65,0: DRAW 0,33
9260 INK 7: BRIGHT 1: PRINT AT 9
    ,12;"Another";AT 10,14;"Go";AT 1
    1,13;"(y/n)"
9270 IF INKEY$="y" THEN RUN
9280 IF INKEY$="n" THEN NEW
9285 GO TO 9270
9500 INK 0: BORDER 7: PAPER 7: B
RIGHT 1: CLS
9510 PLOT 0,0: DRAW 255,0: DRAW
    0,175: DRAW -255,0: DRAW 0,-175
9520 PLOT 239,63: DRAW 0,40: GO
SUB 9800: DRAW -8,0: GO SUB 9800
: DRAW -16,-24: GO SUB 9800: DRA
W -104,-15: GO SUB 9800: DRAW 0,
-2: DRAW 128,0: GO SUB 9800
9530 PLOT 131,67: DRAW 4,4: GO S
UB 9800: DRAW 20,0: GO SUB 9800
9540 PLOT 137,71: DRAW 2,2: GO S
UB 9800: DRAW -12,0: GO SUB 9800
: DRAW 2,2: DRAW 8,0: GO SUB 980
0: DRAW 2,2: GO SUB 9800: DRAW 5
,0: GO SUB 9800: DRAW 2,-6: GO S
UB 9800
9550 PLOT 156,71: DRAW 48,0: GO
SUB 9800: DRAW 7,8: GO SUB 9800
9560 PLOT 120,62: DRAW 114,-4: G
O SUB 9800: DRAW 3,4: GO SUB 980
0
9570 PLOT 237,103: DRAW 0,2: GO
SUB 9800: PLOT 235,103: DRAW 0,2
: GO SUB 9800: PLOT 239,105: DRA
W 0,4: GO SUB 9800: DRAW -8,-2:
GO SUB 9800: DRAW 0,-2: GO SUB 9
800: DRAW 8,0: GO SUB 9800
9580 PLOT 215,71: DRAW 17,7: GO
SUB 9800: DRAW 0,-3: GO SUB 9800
: DRAW -17,-4: GO SUB 9800
9590 PLOT 220,89: DRAW 21,0: GO
SUB 9800: PLOT 221,90: DRAW 21,0
: GO SUB 9800
```

```

9600 PLOT 239,87: DRAW 2,-2: GO
SUB 9800: DRAW 0,-16: GO SUB 980
0: DRAW -2,-2
9610 PRINT AT 4,5;"M U T E   H U
N T E R";AT 8,2;"An Adventure G
ame";AT 10,5;AT 19,6; FLASH 1;"H
IT ANY KEY TO PLAY"; PAPER 0
9620 PLOT 39,144: DRAW 57,0: DRA
W 0,-9: DRAW -57,0: DRAW 0,9: PL
OT 119,144: DRAW 89,0: DRAW 0,-9
: DRAW -89,0: DRAW 0,9
9630 IF INKEY$="" THEN GO TO 96
30
9640 RETURN
9800 BEEP 0.005,30: RETURN
9900 FOR i=1 TO 50: PLOT 128,88:
DRAW 126-INT (RND*240),87-INT (
RND*160): BEEP 0.005,60: NEXT i
9910 LET c$="We have been destro
yed the ": LET c$=c$+e$: LET c$=
c$+" failed Would you like
to try again ? (y/n) ": GO SUB
3000
9920 IF INKEY$="y" THEN GO TO 1
000
9930 IF INKEY$="n" THEN NEW
9940 GO TO 9920
9990 PRINT AT 11,1; INK 7; BRIGH
T 1;"No ";AT 12,1;"Contact"
: RETURN

```

## The Scenario

The year is 2084 and the galaxy is under the direct control of Earth Federation Command (EFC). Since the great galactic war of 2036–54, which established the EFC, there has been peace and the only danger to the galaxy lies in the great drifting wrecked battleships. The occupants of these have mutated over the generations into dangerous creatures of immense power.

This threat has led to the formation of the Mutant Hunter Corps, the sole purpose of which is to seek out and destroy these wandering craft. Each Mute Hunter, as they are called, works alone, transported in his small snub fighter which can travel in

deep space and act as a vehicle for transportation inside the giant ships.

## Instructions

Your mission is to make your way through the battleship XXVII/IV, collecting as many Command Discs as you can, and go to the Teleport. This will transport you into the command area of the ship where you will find an escape pod, which must be marked, and a control panel, which must be set to self-destruct the ship. This may all sound very simple, but remember there are numerous hazards to be found on your travels.

When you start the game, you will be confronted by the control panel of your own ship. In the centre, there is a 3-D image of the corridor in which you are positioned and to the left is a scanner which reports on anything out of the ordinary outside. Your ship is also provided with an instrument status board which will tell you the state of all the onboard systems, and should any of these reach 0 your quest and career will be well and truly over.

## Commands

When you are in COMPUTER mode, you are able to give the following commands:

- OPEN . . . WITH . . . Opens whatever is in front of you.
- TAKE . . . Picks up what is in front of the fighter.
- INVENTORY Gives a list of all items gained during the quest.
- INSERT . . . Inserts item into whatever is in front.
- WAIT Shuts down all systems temporarily.
- MARK Used to mark the escape pod for automatic return after setting the control unit to self-destruct.
- SAVE Saves game at current position.
- LOAD Loads previously stored game.
- CHANGE Changes from COMPUTER into DRIVE mode.

Whenever commands are given to the computer, or received from the computer, they are displayed on the response unit at the top of the screen.

**DRIVE MODE** This mode is used to move around the battleship using stick.1. Pressing the fire button will return to **COMPUTER** mode.

**FIGHTER** If you encounter any hostile beings, then the computer will automatically switch to **WEAPONS** mode for which there is no manual override.

GOOD LUCK AND GOOD HUNTING

# Index

- address book 23-37
  - instructions 34-7
  - programs 23-34
- ADDRESS.1 program 23-7
- ADDRESS.2 program 27-34
- ADVENTURE program 84-95
- adventures 84, 115
- arrays 3, 10
- auto-run facility 4
  
- BASIC 1, 3
- baud rates 97
- birthday book 62
- bits 97
- bytes 97
  
- CATALOG command 2
- central processing unit (CPU) 6-7
- channels 6-9, 99-100
- CLOSE# command 8, 12
- commands 8, 11-12, 97, 148
  - CATALOG 2
  - CLOSE 8, 12
  - ERASE 2-3
  - FORMAT 98
  - GOTO 9999 4, 45
  - IN 114
  - INKEYS# 83, 113
  - LLIST 99
  - LPRIN 99
  - LOAD 4
  - LOAD\* 4
  - MERGE\* 5
  - MOVE 8-9
  - OPEN# 8, 12, 98
  - PRINT# 12, 98
  - SAVE 3
  - SAVE\* 3
  - VERIFY 4
  - VERIFY\* 4
- control unit 6
  
- data base 64
  - updating 66
- data files 8-9, 81-2
  - large 64-70
- data processing 3, 10-22
  - programs 12-22
- data statements 10
- diary, electronic 38-63
- DIARY.1 program 38-42
- DIARY.2 program 42-5
- DIARY.3 program 45-59
- DP Programs 12-22
- DRAW mode 112
- dumping 3
  
- electronic diary 38-63
  - birthday book 62
  - formatting 42
  - initialization 42
  - instructions 59-63
- end of file (EOF) 11, 13
- English/French translator 71-9
  - data base 78-9
  - instructions 78-9
  - programs 71-7
- ERASE command 2-3
  
- fields 10-11
- file-handling programs 12-22
- file management program 17, 27
- files 10-22
- file storage 64
- FORMAT command 98
- FORMAT statement 2
- for-next loop 11
  
- GOTO 9999 command 4, 45
- handshaking 83
  
- IN command 114
- INKEYS# command 83, 113

- joysticks 113-15
- keyboard 8
- key field 11
- LINE statement 3
- LLIST command 99
- LPRIN command 99
- LOAD command 4
- LOAD\* command 4
- memory capacity 64
- MERGE\* command 5
- micro drives 1-5, 64, 82, 105
- micro drive cartridge 2
- MOVE command 8-9
- multiple statements 12
- MUTE HUNTER program 118-48
  - commands 148
  - instructions 148
- network 80-96
  - adventure 84
  - broadcasting 83-4
  - data files 81-2
  - establishing 80
  - programs 81
  - synchronization 83
- NEW/RUN sequence 59-60
- OPEN# command 8, 12, 98
- OPEN# statement 82
- peripherals 6-7, 97, 104
- PHONE file 13-22
- PIE-CHART program 104
- PRINT# command 12, 98
- PRINT# statement 12-13
- printer 8, 104
- programs
  - ADDRESS.1 23-7
  - ADDRESS.2 27-34
  - ADVENTURE 84-95
  - data processing 12-22
  - DIARY.1 38-42
  - DIARY.2 42-5
  - DIARY.3 45-59
  - MUTE HUNTER 118-48
  - PIE CHART 104
  - read only 14-16
  - read/write 17
  - SKETCH 115-18
  - SLIDE.1 105
  - SLIDE.2 106-10
  - TRANSLATOR.1 71-2
  - TRANSLATOR.2 71-7
  - write only 113-14, 23
- read only program 14-16
- read/write program 17
- records 10-11, 35-7
  - changing 37
  - deletion 36-7
- ROM cartridges 113
- RS232 connection 80
- RS232 interface 7-8, 97-104
  - binary channel 99-100
  - commands 97-9
  - control codes 99
  - FORMAT command 98
  - LLIST command 99
  - LPRINT command 99
  - OPEN# command 98
  - PRINT# command 98
  - text channel 99-100
- SAVE command 3
- SAVE\* command 3
- screens 105-6, 111-12
- search and display 36
- serial communication 97
- SKETCH program 115-18
- SLIDE.1 program 105
- SLIDE.2 program 106-10
- slide show 105-12
  - instructions 110-12
- streams 6-9
- synchronization 83
- syntax 3-4
- Tandy four-colour plotter-printer 99-100
- TRANSLATOR.1 program 71-2
- TRANSLATOR.2 program 71-7
- VERIFY command 4
- VERIFY\* command 4
- workspace 64
- write only program 13-14, 23
- ZX-Interface.1 1, 4, 11, 80, 97, 105
- ZX-Interface.2 113-49
  - commands 113
  - instructions 117
  - MUTE HUNTER program 118-48
  - SKETCH program 115-18

**Other titles available from Micro Press:**

**THE MICRO MAZE: A GUIDE TO  
PERSONAL COMPUTING**

Wynford James

0 7447 0000 0

**QUALITY PROGRAMS FOR THE BBC  
MICRO**

Simon

0 7447 0001 9

**QUALITY PROGRAMS FOR THE BBC  
MICRO (Cassette)**

Simon

0 7447 0011 6

**15 GRAPHIC GAMES FOR THE  
SPECTRUM**

Richard G. Hurley

0 7447 0002 7

**20 GAMES FOR THE ORIC-1**

Wynford James

0 7447 0003 5

**MASTERING THE TI-99**

Peter Brooks

0 7447 0008 6

**QUALITY PROGRAMS FOR THE  
ELECTRON**

Simon

0 7447 0004 3

**ADVANCING WITH THE ELECTRON**

Peter Seal

0 7447 0012 4

# MAKING THE MOST OF YOUR SPECTRUM MICRO DRIVES

Having upgraded your Spectrum with the addition of a Micro Drive, you naturally need a comprehensive guide to make full use of this significant expansion of capacity: File Management and Data Processing are covered in detail so that you can write, read and change data addresses efficiently.

An ELECTRONIC DIARY with a daily printout is now well within your Spectrum's capability, and a comprehensive listing for such a diary is included so that you can make, update and delete entries, and automatically receive detailed 'hard copy' of the day's events every morning.

Networking the Spectrum with your friends' machines multiplies the capacity for complex games tenfold, and the author's explanation and listing show you how to exploit this aspect to the full.

## The Author

Richard Hurley read Astrophysics at London University before becoming a maths teacher at Hurstpierpoint College. He is now head of the rapidly developing Computer Studies Department at the College.

GB £ NET +005.95

ISBN 0-7447-0005-1

