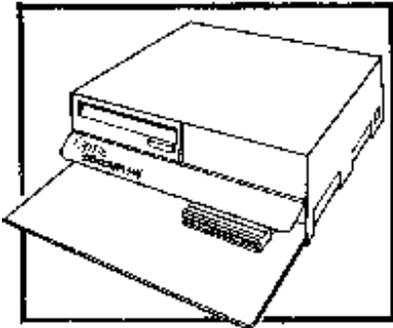


## Discovery disc system operating manual

Congratulations on your purchase of the Opus Discovery disc system which has been specially designed to help you explore the full potential of your Spectrum or Spectrum + computer. Discovery not only complements your computer perfectly but offers you an exciting range of new options including:

- (1) A flexible 3 1/2 inch disc system
- (2) A parallel printer interface
- (3) A joystick interlace compatible with most computer games with a joystick option
- (4) A monitor port
- (5) Peripheral through connector



This user manual will take you, step by step, through the range of new functions offered by Discovery.

If you have not previously used a Spectrum computer we recommend that you first spend a little time reading your Spectrum manual in order to familiarize yourself with the computer's keyboard and the methods of programme entry. After doing this you will be ready to explore the facilities now available to you through your Discovery disc system.

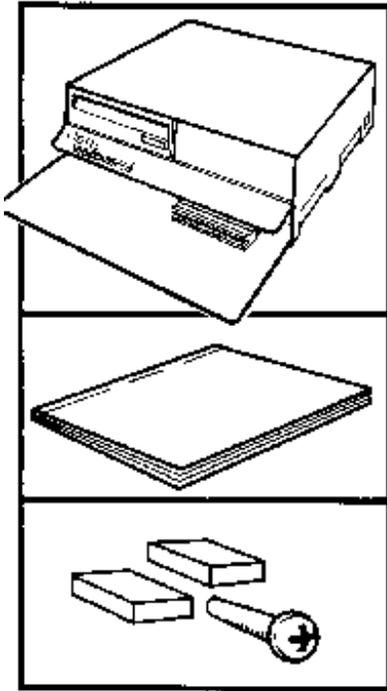
2

## Assembling your Discovery disc system

### Contents

After unpacking your Discovery disc system you will find:

- A) The Discovery disc unit
- B) The user manual
- C) A fixing screw for securing the Spectrum and Discovery 1 unit
- D) Two rubber support 'feet' for the Spectrum



## Assembly

Attaching the Discovery unit to your computer is very simple and preparing your new system for use will take you only a short time.

A) Unplug the power supply from your computer. This will not be needed when using the disc drive system as the Discovery unit has its own power supply.

B) Connect the Spectrum's TV output to your television set using the lead supplied with the computer. Select an unused channel and tune it in to channel 36. If your television has manual tuning you will have to complete your fine tuning after the power has been switched on. Connect the tape drive leads if you will be using a tape based programme.

C) Before connecting the Discovery unit to your Spectrum you will need to study the two diagrams which cover both ZX Spectrum and Spectrum + models.

D) Plug your Discovery unit into the mains supply using the plug already supplied and fitted.

E) Turn on the power supply to your Discovery unit and television set. You are now ready to test the system and the following message should appear on the screen

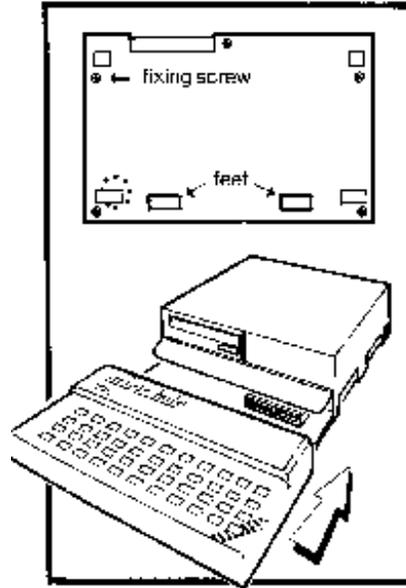
© 1982 SINCLAIR RESEARCH LTD

Some fine tuning may be necessary at this point to produce the best signal and therefore the sharpness of your screen image. If the display fails to appear, turn off the power supply and check all connections before trying again. If the problem continues consult your dealer for further advice.

## ZX Spectrum

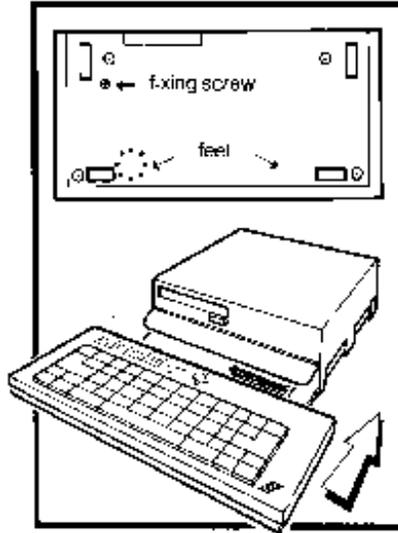
In order to secure the Discovery unit to your ZX Spectrum you will first need to remove the self-tapping screw which is located on the underside of your computer and is arrowed in the

diagram. Now place your computer on the platform at the front of the Discovery unit and align the unit's connector with the connector at the back of your Spectrum. Now slide the two connectors firmly into place. If the connection is complete the hole in the base plate and the vacant screw hole on the underside of your Spectrum should be aligned. Now take the special screw resided with your unit and pass it through the hole provided on the underside of the base plate into the Spectrum. Tighten the screw gently. Fixing is complete when the rubber pad on the base plate touches the computer. Take care not to over-tighten the screw during this operation. Now with the Discovery unit in position take the rubber 'feet' provided and peel off the paper backing. Position the rubber 'feet' on the front underside edge of the computer as indicated in the diagram



### Spectrum +

A hole is already provided on the underside of your Spectrum + computer which will be used in the connection of your Discovery unit: It is arrowed in the diagram. Place your Spectrum + on the platform at the front of the Discovery unit and align the connector with the connector at the rear of your computer. Now slide the two connectors firmly into place. Use the screw provided with your unit to complete the connection. Pass the screw through the hole on the underside of the Discovery platform and into the fixing hole in your Spectrum + : Tighten it carefully. Fixing is complete when the rubber pad on the top of the platform touches the computer. Take care not to overtighten the screw during this operation. Now take the rubber 'feet' provided and peel off the paper backing. Position the 'feet' on the front underside edge of the computer as indicated in the diagram.



4

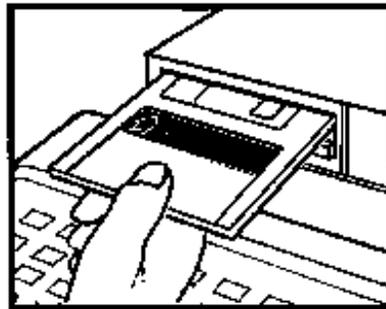
## SECTION 1

### How to start using your disc system

This section of the operating manual has been written for those users with a single drive Discovery unit. Those with a dual drive system will find certain tasks simpler, particularly disc and file copying. Drive 1 is the left hand drive.

To help you distinguish single key commands from those which must be typed in letter by letter all single key entries are marked in *italic*.

You are now ready to start using your Discovery disc system. Take a disc and insert it, as illustrated, into the drive unit, pushing it home until it clicks into place in drive 1.



#### First steps

With the disc in position type the following:

*FORMAT "m" ; 1 ; " First Disc" ENTER*

You have now set up the first disc ready for use and given the disc the name "First Disc". You can easily check that this has been completed by typing:

*CAT 1 ENTER*

This simple command lists the contents of the disc. As there is currently no information on the disc only the title is displayed "First Disc", and the free space on the disc.

## Playing a game

To help you become more familiar with using your new disc system we have devised a simple game for you to play. It's a guessing game. Type:

```
10 LET number = INT (32*RND +1):LET count = 0
20 PRINT AT 2,0;"I have chosen a number between":PRINT AT 3,0;"1 and 32"
30 PRINT AT 5,2;"You have to guess it in the":PRINT AT 6,2;"smallest number of guesses."
40 INPUT "Your guess ";guess
50 IF guess > number THEN PRINT guess;" is too big":LET count = count + 1:GOTO 40
60 IF guess < number THEN PRINT guess;" is too small":LET count = count + 1:GOTO 40
70 PRINT guess;" is the correct answer."""It took you ";count+1;" guesses."
```

Now just before you try the game we can show you very quickly how to save it on disc by typing:

```
SAVE * "m";1;"guess" ENTER
```

Again you can check whether it has worked by typing:

*CAT 1 ENTER*

The programme 'guess' should now appear in the list of disc contents. You have just learnt how easy it is to save your programme on disc. Now try the game by simply pressing:

*RUN ENTER*

If the programme fails to run you may have made a programme entry mistake. Retype any incorrect lines and then type:

```
SAVE * "m";1;"guess" ENTER
```

This will save the correct version and automatically replace the incorrect programme. Now try the game again.

## 5

Now type:

*NEW ENTER*

This removes the programme. Don't worry, the programme is still stored on your disc so you can bring it back whenever you want. Instead of entering the game programme all over again all you have to do is to type:

```
LOAD * "m";1;"guess" ENTER
```

To make sure the programme is back,type:

*LIST ENTER*

The programme will be displayed and can be run by typing:

*RUN ENTER* as before.

## Summary

To clear the 'guess' programme and then recover it, use the following sequence:

*NEW ENTER*

*LOAD \* "m";1;"guess" ENTER*

*LIST ENTER*

*RUN ENTER*

If the message "file not found" should appear check your entry again.

Remember to type in your file name exactly as your original. In this case "guess" is in lower case. "GUESS" entered in upper case would produce "file not found".

## Automatic run facility

Another useful feature of your Discovery disc system is its ability to run a programme automatically.

Type:

*NEW ENTER*

Now type the following programme:

```
10 BORDER 0:PAPER 0:INK 7:CLS :PRINT AT 10,7;  
20 FOR i=1 TO 17:READ a:PRINT CHR$ a;:NEXT i  
30 DATA 73,116,39,115,32,97,117,116,111,109,97,116,105,99,32,33
```

Save the programme on disc by typing:

*SAVE \* " m";1;"run" LINE 10 ENTER*

Now type:

*NEW ENTER*

*RUN ENTER*

The programme runs automatically from line number 10. Typing *RUN* after typing *NEW*, or after first switching on ,causes the programme you have selected to call 'run' to run automatically. If you find a favourite programme you may find it useful to name the programme 'run' so that you can take advantage of the automatic run facility.

## Using disc based programmes

Games and other programmes supplied on disc will normally use the automatic run facility just

outlined. All you need to do is to turn on your system, insert your disc and type:

*RUN ENTER*

The programme runs automatically. If this method is not used, follow the instructions supplied with the disc programme.

### Using tape based programmes

To use tape based programmes the Spectrum must be connected to a suitable tape recorder using the lead supplied with the computer. Connect the MIC and EAR sockets on the Spectrum to the corresponding sockets on the tape recorder and set the volume control to a high setting. Full details on entering tape based programmes are contained in your Spectrum computer operating manual.

### Transferring programmes from tape to disc

To transfer a programme called 'IMPACT' (for example) from tape to disc, you will first need to load it using:

*LOAD "IMPACT" ENTER*

## 6

Let the programme finish loading then type:

*SAVE \*"m";1;"IMPACT" ENTER*

For more information on transferring programmes from tape to disc see appendix 1 .

### Using joysticks with Discovery 1

When playing computer games you may often need to use five or more different keys. This can make some games very difficult which is why many games now offer a joystick option. The Discovery disc system has a joystick interface and is compatible with most games which allow joysticks to be used. To use a joystick with a suitable programme plug the joystick into the socket illustrated.



Now you have to use a command to turn the joystick on. Type:

*FORMAT "j";1 ENTER*

You can now run the game programme. When you have finished you can turn off the joystick by typing:

*FORMAT "j";0 ENTER*

You can also use the joystick in your own programmes. We have devised a programme to show you how. Type the following programme:

```
10 CLS :PRINT AT 4,0;" Use the joystick to draw a":PRINT AT 5,0;"picture"  
20 PRINT AT 7,0;"Use the fire button to end"  
30 PRINT AT 20,1;"Press any key when ready"  
40 LET k$=INKEY$:IF k$="" THEN GOTO 40  
50 FORMAT "j";1  
110 BORDER 0:PAPER 0:INK 7:CLS  
120 PLOT 0,0:DRAW 255,0:DRAW 0,175:DRAW -255,0:DRAW 0,-175  
130 LET stick = IN 31:IF stick = 0 THEN GOTO 130  
140 IF stick = 1 THEN DRAW 1,0  
150 IF stick = 2 THEN DRAW -1,0  
160 IF stick = 4 THEN DRAW 0,-1  
170 IF stick = 8 THEN DRAW 0,1  
180 IF stick >= 16 THEN FORMAT "j";0:STOP  
190 GOTO 130
```

The command 'IN 31 ' on line 130 reads the position of the joystick and this is used to decide where to move next.

You can save the programme with the name 'joysketch' on disc by using the command:

*SAVE \*"m";1;"joysketch" ENTER*

As this joystick programme has probably taken you some time to input into your computer it would be a pity to lose it. You can check that the whole programme has been correctly saved by typing:

*VERIFY \*"m";1;"joysketch" ENTER*

This command checks (verifies) that what you have saved on disc is the same as the programme in the computer's memory and gives you an error message if it is not.

Having verified that the programme is saved correctly you can now run it by typing:

*RUN ENTER*

When you have a sketch that you particularly like, stop the programme by pressing the joystick's fire button. You can even save the picture on disc by typing:

*SAVE \*"m";1;"sketch" SCREEN\$ ENTER*

(Spectrum + owners have the abbreviated single key command *SCRN\$*)

Now to clear and reset the screen type:

*CLS # ENTER*

You can easily restore the picture by typing:

*LOAD \*"m";1;"sketch" SCREEN\$ ENTER*

(SCRNS)

7

After a few seconds the picture will appear as before.

## Data handling

As you are about to discover your computer is not only useful for playing games but also for storing information.

The following programme will show you how. Type:

```
10 OPEN # 4,"m";1;"telephone" OUT
100 INPUT "Name?";n$,"Tel.No.?";t$
110 IF n$="END" OR n$="end" THEN CLOSE #4; STOP
120 PRINT #4;n$
130 PRINT #4;t$
140 GOTO 100
```

Save the programme with the name 'tel1' by using the command:

```
SAVE *"m";1;"tel1" ENTER
```

Now run the programme. If the programme fails to run, check your entries, correct them and save the programme again as outlined earlier.

Type in the names and telephone numbers of a few friends and complete your list with 'end' and any telephone number. The programme will save the names and telephone numbers in the file 'telephone' and they can be listed whenever you require them. Now type in the following programme:

```
10 OPEN # 4,"m";1;"telephone" IN
100 PRINT #4;
110 IF USR 432=0 THEN CLOSE # 4:STOP
120 INPUT #4;n$;t$
130 PRINT n$,t$
140 GOTO 100
```

Save this programme as 'tel2' by using the command:

```
SAVE *"m";1;"tel2" ENTER
```

When you run this programme it will list the names and addresses previously saved in the file 'telephone'.

The programme above stores the names and telephone numbers one after another and to find a particular name and number all of the entries have to be shown. Obviously it would be better if you could pick out the name and telephone number without having to read through all the entries in your file. The Discovery disc system allows information to be stored in this way and an example of how to do this is given later in the section on Random Access Files.

## Disc care and copying

The Discovery Disc system uses 3 1/2 inch discs contained within a durable hard case: Even so the

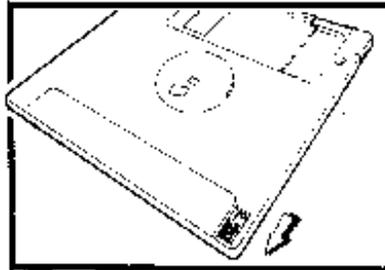
following simple rules should be followed to minimise problems:

Do not touch exposed surfaces

Keep away from magnetic fiends generated by televisions, tape recorders,loudspeakers etc.

Avoid dust, cigarette ash or liquids. If the surface should become contaminated do not use the disc again as the contaminant will be transferred to the disc drive and may then contaminate other discs.

Important discs can be 'write protected' so that they cannot be written to by the disc system: To do this hold the disc as if you were going to insert it into the disc drive. Underneath the disc on the left hand side you will find a slide covering a hole in the disc case. When the hole is covered the disc can be written to. Sliding the cover back to uncover the hole prevents the disc system writing to the disc and the disc is therefore 'Write protected'. As a further precaution you should keep copies of discs containing important information.



## Disc copying

To make a copy of a disc first format a new disc as already described in the operating manual. Now remove the disc and insert the disc to be copied and type:

8

```
MOVE "d";1 TO "d";3 ENTER
```

From this point on the programme will prompt you when it is necessary to swop the discs and complete the copying procedure .Be careful not to mix up the source and destination discs - your original and your copy. It is a good idea to write protect the source disc (your original) before starting the copying procedure. For those with two disc drives the disc swapping can be avoided by copying from one drive to the other. To copy the disc in drive number one onto a formatted disc in drive number two type:

```
MOVE "d";1 TO "d";2 ENTER
```

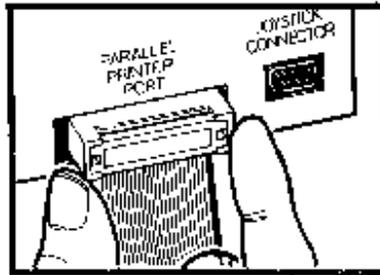
The disc will be copied automatically.NOTE: When the disc is copied it is also compacted. See the later section on compacting a disc.

## Using a printer with your discovery system

The Discovery disc system can be used with a parallel printer. To do this you will need a Discovery printer cable which can be obtained from your dealer. The cable has a Centronics printer connector at one end and an edge connector for the Discovery disc unit at the other. It also has a red strip along one edge which is used to ensure that the edge connector is correctly inserted.

To connect the Discovery unit and the printer, first turn off the power to both units. The edge connector for the

Discovery unit should be inserted into the printer port located on the right hand side of the unit with the red stripe towards the rear.



The other end of the cable should be connected to the printer as outlined in the printer manual. Now turn the power on again and type:

```
OPEN # 3;"t" ENTER
LPRINT "Printertest -abcdefghijklmnopqrstuvwxyz" ENTER
CLOSE # 3 ENTER
```

The line "Printertest - abcde etc' should have been printed on your printer. If not turn off the power and check all the connections before trying again. Once the printer is working it can be used to list programmes and to print other information. Load one of the previous example programmes and then type:

```
OPEN # 3;"t" ENTER
LLIST ENTER
```

The programme will be listed on the printer. Now type in the following programme:

```
10 OPEN # 4;"t"
20 PRINT #4; TAB 3;"squares"
30 FOR 1 = 1 TO 20
40 PRINT # 4;i; " " ;i * i
50 NEXT i
60 CLOSE # 4
```

When this programme is run the output, a table of squares, will be printed on the printer.

## Information for microdrive users

Except for the differences listed here the Discovery system is completely software compatible at

command line level with the microdrive system. BASIC programmes which work on the microdrive system should work unchanged.

The differences between the Discovery disc system and the microdrive system are:

*SAVE \* "m";1;"fred"* replaces the programme 'fred' if it already exists.

*INKEY\$ # 4* returns a null string when the end of file is reached.

## 9

The list of files produced by the *CAT* command is not in alphabetical order.

You can *MERGE \** programme files that have been saved using the *LINE* option.

The separator in the *OPEN #* command can be a semi-colon or a comma, even for the channels "k", "s" and "p".

Also some of the commands have been extended and these extensions are explained in the appropriate sections.

The Discovery disc system and the microdrive system cannot be used together and to transfer programmes from microdrive to disc they will first have to be saved on tape. They can then be loaded from tape and saved onto disc.

Programmes supplied on tape with instructions to convert them for use on the microdrive system should convert to disc in exactly the same way.

## Section 2

This section of the user manual covers the new commands of the Discovery disc system in more detail. It uses the programmes and data from the introductory section in some of the examples given to illustrate the commands. You should therefore work through the introductory section first and build up your own copy of the 'First Disc'.

If you have entered and run a programme which does not stop normally - in other words you find your computer has

become stuck in a 'programme loop' - you can interrupt the programme by pressing the *CAPS SHIFT* and *SPACE* keys

together. (Spectrum + users need only press the *BREAK* key).

Be careful however to avoid doing this when a disc drive is being used as it may corrupt the information on the disc.

Always try to make copies of important discs to safeguard your information.

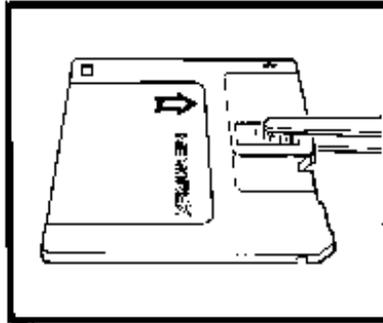
### Setting up the system -

*FORMAT* and *CAT*

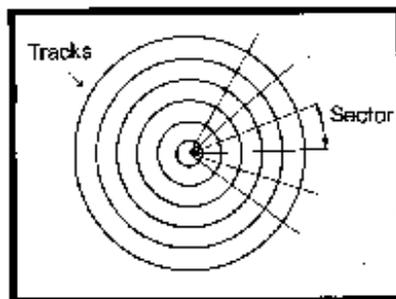
The Discovery disc system consists of the hardware and software necessary to enable your Spectrum to transfer

information to and from a disc, together with a number of extra facilities. The system allows one or two disc drives to be used. For those with a dual drive unit, the left-hand drive is drive number 1. The 3 1/2 inch discs used with the Discovery disc system have a hard case and are particularly robust. Inside the case is a circular, flexible disc with a magnetic coating. When inserted into the

drive the protective shutter is moved aside giving access to the recording surface. When reading (or writing) the disc head moves across the required position and the information is read (or written) as the disc rotates under the head. Unlike a record the information is not recorded on a continuous spiral path, but, on a series of concentric circles. The disc head can be moved in and out across the surface to a number of fixed positions. Once in position the head traces out a circular path, or track, on the disc as it rotates. The number of tracks on the 3 1/2 inch discs used in the Discovery system is 40.



Each track holds a large amount of data and is therefore further divided into a number of equal parts, called sectors. On the Discovery disc system each track is divided into 18 of these sectors.



## 10

Each sector holds 256 bytes and so a forty track disc will hold up to 180 Kbytes (1 Kbyte being 1024 bytes).

When you purchase a disc there is nothing recorded on it. But before you can use it it must be set up. This process is called formatting the disc and is done by using the *FORMAT* command, for example:

```
FORMAT "m";1;"Disc Two" ENTER
```

formats a disc in drive number one and gives it the title 'Disc Two'. Formatting a disc erases any information that was stored upon it (you may be re-using an old disc) and so you must be careful not to format any disc containing information that you want to keep. Discovery however has been designed to prevent this happening. If you try to format a disc that has already been formatted and given the name 'utilities' (for example), you will be asked:

Destroy "utilities"?

Then if you wish to proceed you should type 'y' to format the disc. The disc title enables you to identify your discs and can be up to ten characters long. For identification purposes each sector on the disc has an identification marker.

These markers are not written along with the information when it is saved but must be written before the disc can be used. Formatting is the process of writing these markers. When information is stored in a filing cabinet it is kept in cardboard folders which are labelled to allow a particular folder to be found easily. In a similar way information on the disc is stored in 'files' and each one is given a name. The index of all files on the disc (showing where they are located) is stored in a special file called the catalogue file. To display the catalogue file of a disc the *CAT* command is used e.g.:

*CAT 1 ENTER*

This displays the catalogue of the disc in drive number 1 : The catalogue of a newly formatted disc has the title that you specified when it was formatted and has no files present.-The number shown at the end of the list of files is the amount of space remaining on the disc (in kilobytes- 1024 bytes).

The catalogue contains the name and location of each file on the disc together with other information used by the disc operating system. The catalogue is itself a file and this means that there is no restriction on the number of files allowed except the space available on the disc itself. See the reference section for further details of the catalogue file.

## The *MOVE* command

One of the simplest things that you can do with your Spectrum is to take letters from the keyboard and display them on the screen. This can be done by typing:

*MOVE "K" TO "S" ENTER*

Now type in some letters, the letters that you have typed will appear on the screen. To finish, press *CAPS SHIFT* and *ENTER* together. The *MOVE* command takes input from the keyboard (identified by the "K") and transfers it to the screen (identified as "S"). In this example, and when working through the introductory section, you will have noticed that different parts of the system are identified by a single letter in quotes. For example "K" for the keyboard, "S" for the screen or "t" for the parallel printer. These different parts of the computer system are called channels

Now try this:

*MOVE "K" TO "K" ENTER*

Again type in some letters and this time the letters appear on the bottom part of the screen. To finish, press *CAPS SHIFT* and *ENTER* together. The 'K' channel can be used for input (from the keyboard) and for output (to the lower part of the screen).

The examples just shown are of little practical use however and so let's try a number of more useful examples.

When you have run the programme 'Tel1' from the introductory section you can display the contents of the datafile

'telephone' by putting the disc into drive number 1 and typing:

*MOVE "m";1;"telephone" TO "S" ENTER*

The contents of the file will be displayed on the screen. Channel "m" is the disc file channel.

## 11

You also have to tell the system which disc drive is being used and which file you require. "m";1;"telephone" refers to the file 'telephone' on the disc in drive number one. The command you entered transfers the contents of the file to the screen. This time there is no need to indicate that the input has finished as this is done

automatically when the end of the file is reached.

The output obtained by using the *MOVE* command is not as tidy as that obtained by using the programme 'tel2'. The *MOVE* command just takes characters from the file and puts them onto the screen. It does not check what is being sent and makes no attempt to obtain a tidy display. This is further illustrated by the next example.

Try to display the programme 'tel1 ' by using the *MOVE* command:

*MOVE "m";1;"tel 1" TO "S" ENTER*

this does not provide a listing of the programme because programmes are not stored in the same way as data. The

*MOVE* command can only be used therefore to display data files. In the same way the *MOVE* command can

also be used to print a data file using the parallel printer interface by typing, for example:

*MOVE "m";1;"telephone" TO "t" ENTER*

this will print the contents of the data file 'telephone' (on the disc in drive number one) on a parallel printer connected to the parallel port. When used for output the "t" channel sends printable characters to the parallel port and filters out other characters. As with the screen display, this will not work sensibly with programme files.

Those with a dual drive system can make a copy of the file 'telephone' from a disc in drive number one onto a formatted disc number two by typing:

*MOVE "m";1;"telephone" TO "m";2;"telephone" ENTER*

Those with a single drive system can make a copy by using the drive numbers one and three, in other words by typing:

*MOVE "m";1;"telephone" TO "m";3;"telephone" ENTER*

As the file is transferred you will be prompted to change discs when necessary.

Drive numbers one and three both refer to the physical drive number one. When you change from using drive number one to using drive number three (and vice versa) the disc system recognises the need to change discs and prompts you to do so. The same situation applies with drive numbers two and four, they both refer to physical drive number two.

Programme files can be moved in exactly the same way, 'tel1' can be copied by typing:

*MOVE "m";1;"tel1" TO "m";3;"tel1" ENTER*

The disc system uses the *MOVE* command for a number of special functions - disc copying, disc compacting and changing a file's position on the disc. Disc copying and compacting are described elsewhere but if you want to move the location of 'file 1' on the disc use the command:

```
MOVE "m";1;"file1" TO "m";1;"file1"
```

This can be useful if when trying to extend 'file1' you obtain a file size error. This indicates that there is insufficient space at the current location to extend the file.

The *MOVE* command transfers data from an input channel to an output channel.

Information on the channels available to you is given in the later section on channels and streams. One final example:

The 'b' channel allows input and output using the parallel port, without translation (in contrast to the 't' channel). This can be used if you want to transfer information to or from another Spectrum. You can even do this with other computers (providing suitable software is available).

To send a file, 'sfile' on disc number one, use the command:

```
MOVE "m";1;"sfile" TO "b"
```

To receive information and save it in the file 'rfile' use the command:

```
MOVE "b" TO "m";1;"rfile"
```

the computer receiving the information should carry out its commands first so that it is ready to receive when the machine sending the information starts.

## 12

When the transfer is complete the person receiving should generate an end of file by pressing *CAPS SHIFT* and *ENTER* together. This writes any remaining information to the disc. To transfer programme files it is better to use *LOAD \** and we will explain how to do this in the next section.

### Load \* Save \* Merge \* and Verify \*

These commands have the same effect as the corresponding cassette commands. This section contains a number of further examples to illustrate the use of these commands.

Insert your copy of the 'First Disc' into drive number one and type:

```
LOAD *"m";1;"guess" ENTER
```

This will load the file called 'guess' from the disc in drive number one. To save the programme on another formatted disc insert the new disc into drive no. 1 and type:

```
SAVE *"m";1;"guess" ENTER
```

When loading a file, if the file does not exist an error report 'file not found' will be generated. When saving a file, if the file does not exist then a new file is created and the programme or other information is saved to that file. If the file already exists then, unlike the microdrive system, it will be overwritten.

*LOAD \** and *SAVE \** can also be used to save the contents of the screen by using the *SCREEN\$* option. Load the programme 'joysketch' and modify it by typing:

```
180 IF stick >= 16 THEN SAVE * "m";1;"sketcha" SCREEN$ :FORMAT "j";0:STOP ENTER
```

Now save this new version as 'joysketcha' by typing:

```
SAVE *"m";1;"joysketcha" ENTER
```

Now run the programme. When your picture is finished press the fire button and it will automatically be saved. You can then reset the screen and reload the picture by typing:

```
CLS # ENTER  
LOAD *"m";1;"sketcha" SCREEN$ ENTER
```

*CLS #* clears the screen and resets the screen attributes to their default values (these are given in the reference section).

Another extension to the *LOAD \** and *SAVE \** commands allows an array to be saved and loaded.

The following

programme generates 100 random numbers between 1 and 10 storing them in the array *r*. This array is then saved on

disc as the file 'rand.no.s'. Type:

```
10 DIM r(100)  
20 FOR i=1 TO 100  
30 LET r(i)=1 + INT (RND * 10)  
40 NEXT i  
50 SAVE *"m";1;"rand.no.s" DATA r()
```

Save the programme on disc with the name 'rgen' using the command:

```
SAVE *"m";1;"rgen" ENTER
```

Run the programme and then erase the programme and variables by typing:

```
NEW ENTER
```

The array can be reloaded by using the corresponding *LOAD \** command. Type in the following programme:

```
10 DIM c(100)  
20 LOAD *"m";1;"rand.no.s" DATA c()  
30 FOR k=1 TO 100  
40 PRINT c(k);" "  
50 NEXT k  
60 PRINT
```

When run this programme loads and prints the random numbers saved by the programme 'rgen'.

Note that the array

name does not have to be the same in both programmes, but that the array size (100 numbers) does.

This extension can

be used with string arrays as well as with arrays of numbers. *LOAD \** and *SAVE \** (can also be used to save machine code programmes by using the *CODE* option. Type:

```
SAVE *"m";1;'mcode' CODE 32500,67 ENTER
```

this will save the 67 bytes of code starting from memory location 32500. This can be reloaded when required by using the corresponding load command:

```
LOAD *"m";1;"mcode" CODE ENTER
```

### 13

this loads the machine code at the same address as when it was saved. To load at a specified address (which may be different) use the command:

```
LOAD *"m";1;"mcode" CODE 60000 ENTER
```

This command loads the machine code starting at 60000. It does not check that the space which you have allowed for the machine code is large enough. If you have set aside 100 bytes for a machine code programme you can make sure that other memory is not overwritten by specifying the length in the load command, for example:

```
LOAD *"m";1;"mcode" CODE 60000, 100 ENTER
```

If the code in the file 'mcode' is larger than 100 bytes an error report will be generated. Using *CODE* any block of memory can be saved, it does not have to be machine code. To save the screen for example the command:

```
SAVE *"m";1;"screen" CODE 16384,6912 ENTER
```

could be used, although it is simpler to use the *SCREEN\$* option described earlier. This can then be redisplayed by using the corresponding *LOAD \** command:

```
LOAD *"m";1;"screen" CODE 16384,6912 ENTER
```

After saving a programme or other information using the *SAVE \** command you can check that it has been saved correctly by using the *VERIFY \** command. All of the extensions, like *CODE* can be used with this command.

Load the programme 'joysketcha' and save it on another formatted disk by typing:

```
SAVE *"m";1;"joysketcha" ENTER
```

You can check that it has been saved correctly by typing:

```
VERIFY *"m";1;"joysketcha" ENTER
```

any difference between what is saved and what is in memory will cause an error report to be generated. Now run the

programme. When ready to save a picture, press the fire button. The picture which you have chosen to save can be checked in the same way by typing:

```
VERIFY *"m";1;"sketcha" SCREEN$ ENTER
```

any discrepancy will produce an error report.

The *MERGE* \* command combines a programme and variables from disc with one already in memory. Any programme lines, or variables that are duplicated are overwritten. To illustrate this, type in the following programme lines:

```
140 IF stick = 1 THEN DRAW 1,0:GOTO 130
150 IF stick = 2 THEN DRAW -1,0:GOTO 130
160 IF stick = 4 THEN DRAW 0,-1:GOTO 130
170 IF stick = 5 THEN DRAW 1,-1:GOTO 130
180 IF stick = 6 THEN DRAW -1,-1:GOTO 130
190 IF stick = 8 THEN DRAW 0,1:GOTO 130
200 IF stick = 9 THEN DRAW 1,1:GOTO 130
210 IF stick = 10 THEN DRAW -1,1:GOTO 130
220 IF stick < 16 THEN GOTO 130
230 SAVE *"m";1;"sketchb" SCREEN$
240 FORMAT "j";0:STOP
```

Save this as the file 'updatejoy' by typing:

```
SAVE *"m";1;"updatejoy" ENTER
```

Now load the programme 'joysketch' and list it by typing:

```
LOAD *"m";1;"joysketch" ENTER
LIST ENTER
```

The new programme lines can be merged from the file 'updatejoy' by typing:

```
MERGE *"m";1;"updatejoy" ENTER
```

Now list the programme and you should find that the programme lines you entered have been combined with the existing programme. Lines 140-190 of the original programme have been overwritten while lines 200-240 have been added. The changes allow the programme to detect other positions of the joystick and to move accordingly. This programme is further explained in the next section of the manual.

## 14

To save the programme on disc so that it will run automatically type:

```
SAVE *"m";1;"joysketchb" LINE 10 ENTER
```

Use of the *LINE* option to the *SAVE* \* command means that when loaded again the programme will

run automatically from line number 10. First check that the programme has been saved correctly by using *VERIFY \** i.e. by typing:

```
VERIFY * "m";1;"joysketchb" ENTER
```

Provided that this is successful, remove the programme by typing:

```
NEW ENTER
```

Now load the programme again by typing:

```
LOAD *"m";1;"joysketchb" ENTER
```

The programme should run automatically.

The *MERGE \** command can also be used to load an autorun programme without making it run. Type:

```
NEW ENTER
```

```
MERGE *"m ";1;"joysketchb" ENTER
```

Any programme and variables are removed by the *NEW* command. As there is no programme in memory, merging

'joysketchb' has the same effect as loading, except that the programme does not run automatically. All the commands just described can also be used with the 'b' channel. This channel sends and receives information from the parallel port. Unlike the 't' channel, characters are sent and received with no translation. This channel can therefore be used, with a suitable connecting cable, for the transfer of programmes from one Spectrum to another. The receiver should type:

```
LOAD *"b" ENTER
```

and then after loading the programme the sender types:

```
SAVE * "b" ENTER
```

Once the transfer is completed it can be checked using the *VERIFY \** command. The receiver types:

```
VERIFY * "b" ENTER
```

and the sender 'saves' the programme again. The programme is now in the receiver's memory and so the *VERIFY \**

command checks this against what is sent the second time. Once transferred the programme can be saved to disc in the

normal way. This is much better than using the *MOVE* command as it allows the information sent to be verified and does not need an end of file to be generated from the keyboard.

## The joystick programme

Load the 'joysketchb' programme without running it by using the *MERGE \** routine described earlier.

*LIST* the programme.

```
10 CLS :PRINT AT 4,0;" Use the joystick to draw a":PRINT AT 5,0;"picture"
```

```

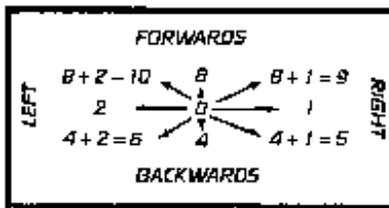
20 PRINT AT 7,0;"Use the fire button to end"
30 PRINT AT 20,1;"Press any key when ready"
40 LET k$=INKEY$:IF k$="" THEN GOTO 40
50 FORMAT "j";1
110 BORDER 0:PAPER 0:INK 7:CLS
120 PLOT 0,0:DRAW 255,0:DRAW 0,175:DRAW -255,0:DRAW 0,-175
130 LET stick = IN 31:IF stick = 0 THEN GOTO 130
140 IF stick = 1 THEN DRAW 1,0:GOTO 130
150 IF stick = 2 THEN DRAW -1,0:GOTO 130
160 IF stick = 4 THEN DRAW 0,-1:GOTO 130
170 IF stick = 5 THEN DRAW 1,-1:GOTO 130
180 IF stick = 6 THEN DRAW -1,-1:GOTO 130
190 IF stick = 8 THEN DRAW 0,1:GOTO 130
200 IF stick = 9 THEN DRAW 1,1:GOTO 130
210 IF stick = 10 THEN DRAW -1,1:GOTO 130
220 IF stick < 16 THEN GOTO 130
230 SAVE *"m";1;"sketchb" SCREEN$
240 FORMAT "j";0:STOP

```

Line number 50 uses the *FORMAT* command to turn the joystick on, while line number 240 turns the joystick off and then stops the programme. The 'j' channel is only used to turn the joystick on and off using the *FORMAT* command. 1 turns the joystick on and 0 turns it off.

## 15

Line number 130 uses *IN 31* to obtain the position of the joystick. If the joystick is not being moved a value of 0 is returned and this means that the programme jumps back to get another value. The programme stays in this loop until a non zero value is obtained. Line numbers 140-210 test the value of 'stick' and move accordingly. The relationship between positions of the joystick and values of 'stick' obtained are shown in the following diagram:



Pressing the fire button adds 16 to the value of 'stick'. Line number 220 tests if the fire button is pressed or not. If it is not pressed the condition 'stick < 16' will be true and the programme jumps back to line 130. If the button is pressed the programme continues with line number 230 and saves the screen, turns the joystick off and then stops.

## Channels and streams

In earlier sections of the manual we have seen that the Spectrum identifies different parts of the system by using a single letter in double quotes. These different items are called channels. The main channels available on the Discovery system are:

"K" - Input from the keyboard and output to the lower part of the Screen.

"S" - Output to the upper part of the Screen .

"P" - Output to the ZX printer.

"m" - Input or output to a disc file.

"b" - Input or output to the parallel port.

"t" - Input or output to the parallel port.

### Note 1)

The 'm' channel needs further information to specify it completely. For example:

"m";1;"data"

refers to the file 'data' on the disc in drive number 1 .

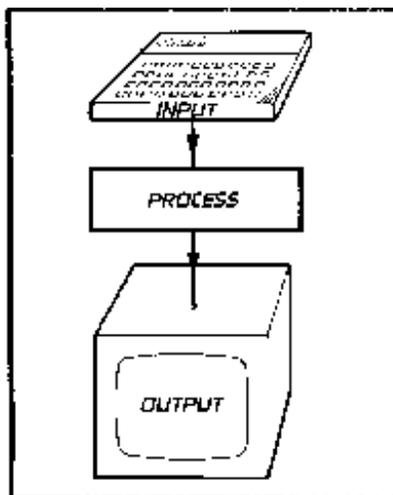
### Note 2)

The 'b' channel sends and receives characters without changing them in any way. The 't' channel does make changes to the characters. For output, BASIC tokens are expanded and graphics characters are printed as '?' (see the reference section for details). For input characters have their most significant bit set to zero. The 't' channel is used as a text channel, while the 'b' channel is used as a binary channel.

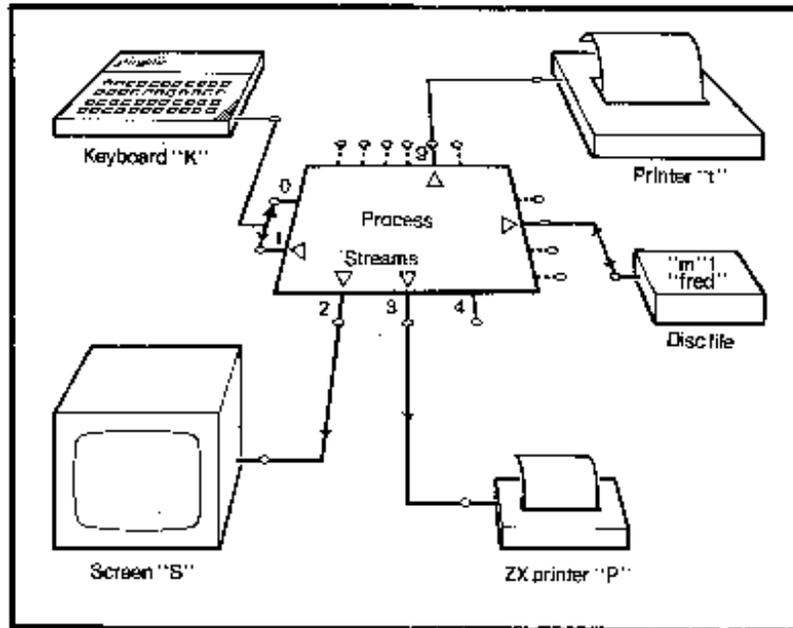
### Note 3)

Other special channels are available. These have been used in the examples and are described in the reference section.

Data is information in a form that can be processed by the computer and can be said to flow from input to output.



Using this analogy, information moving around the system is called a data stream.  
 At any instant there may be several streams of data in existence. Each data stream is identified by a number from 0 to 15 and is 'linked' to various input or output devices (channels).



Stream numbers 0,1 and 2 are already assigned and are best left alone, at least until you become totally familiar with your system. The assignments for streams 0,1 and 2 are:

### 0 and 1

These are used for input from the keyboard and for output to the lower half of the screen. Used automatically by *INPUT* and *INKEY\$* ( not *INPUT #* or *INKEY\$ #* these use the specified streams).

### 2

Is used for output to the upper part of the screen. Used automatically by *PRINT* and *LIST*.

### 3

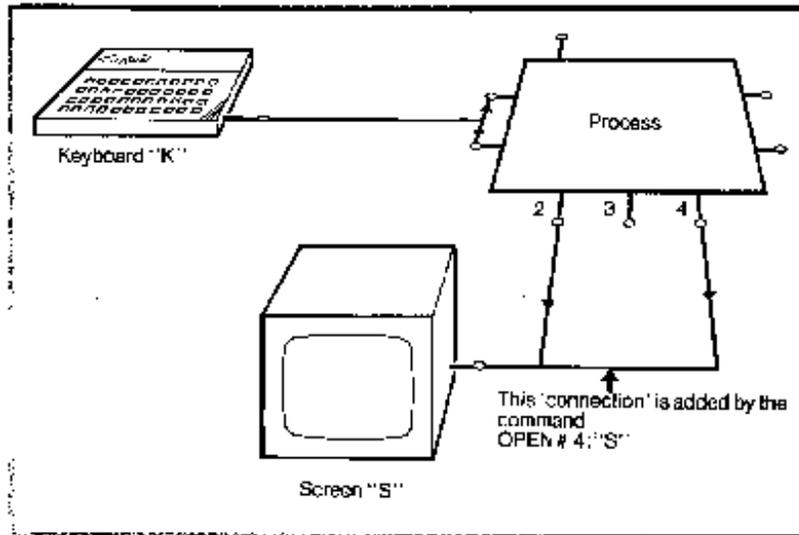
Stream 3 is automatically assigned to the printer but it can be reassigned if you wish. This is useful if you wish to redirect output to the printer as Stream 3 is automatically used by *LPRINT* (not by *LPRINT #* which uses the specified stream) and *LLIST*

In BASIC streams and channels are 'linked' by using the *OPEN #* command, for example:

17

```
OPEN # 4;"S" ENTER
```

associates stream number 4, with the screen, channel "S" .



The *OPEN #* command makes the connection between stream four and the actual channel device "S". To break the connection and allow stream four to be used for something else you must use the following command:

*CLOSE # 4 ENTER*

Stream four can now be linked to another channel, for example:

*OPEN #4;"t" ENTER*

now associates stream four with the parallel port.

Type in the following programme which takes input from the keyboard and displays it on the screen in a similar way to the *MOVE* command.:

*MOVE "K" TO "S"*

```
30 LET a$ = INKEYS : IF a$ = "" THEN GOTO 30
40 PRINT a$;
```

Run the programme. Characters typed on the keyboard are printed on the screen. This time to stop the programme press *BREAK* or *SHIFT* and *SPACE* together. As written the programme uses the default channels. Now type in the following programme:

```
10 OPEN #4;"K"
20 OPEN # 5; "S"
30 LET a$ = INKEY$ # 4: IF a$ = "" THEN GOTO 30
40 PRINT # 5;a$;
```

This programme works in exactly the same way, but, it now uses stream 4 for input and stream 5 for output. This programme is far from perfect. To see why, run the programme and after typing in a few characters stop it by pressing *CAPS SHIFT* and *SPACE* together (*BREAK*). Now try to run the programme again. It will fail with

the error report 'stream already open '. This happened because the programme did not close the streams it was using with the *CLOSE #* command.

## 18

The only way of stopping the programme is to break into it and this leaves the stream open. This is poor programming practice and should be avoided. If you have to break into a programme, or if the programme fails, any open streams can be reset by using the *CLEAR #* command. Type in the command:

```
CLEAR # ENTER
```

Now run the programme again. This time it should run as before. Using *CLEAR #* to reset a stream used for output to a file channel may mean that part of the information is lost. Closing a stream linked to an output file writes any remaining information to the disc. This is not done by the *CLEAR #* command which just resets the streams and makes them available again. If you wish, you can clear single stream, the command:

```
CLEAR # 5
```

just resets stream five.

The *OPEN #* command links a specified channel device to the given stream. You can specify whether the channel is to be used for input or output by using the *OPEN #* command, for example if you type:

```
OPEN # 7;"b" IN
```

you will link stream 7 to the "b" channel for input only. Similarly the command:

```
OPEN # 5;"m";1;"test" OUT
```

will link stream 5 to the disc file 'test' on drive number one. The file is to be used for output. Disc file channels have a number of other options and these are described in the next section on data storage. As the examples just shown illustrate, the type of use may be omitted from the *OPEN #* command and the option that is then taken depends upon the channel being used. Details on the use of channels is given in the reference section.

## Data storage

To explain data storage more fully let's return to the programme 'tell' which we first used in the introductory section.

Load and list the programme to display:

```
10 OPEN # 4;"m";1;"telephone" OUT  
100 INPUT "Name ? ";n$,"TeI. .No ?";t$  
110 IF n$ = "END" OR n$ = "end" THEN CLOSE # 4:STOP  
120 PRINT #4;n$  
130 PRINT # 4;t$  
140 GOTO 100
```

Line 10 opens a file 'telephone' on the disc in drive number one for output. This file output channel is 'linked' to stream number four. The *OUT* part of the *OPEN #* command indicates that the file is to be used for output. If the file does not exist then a new file will be created, if the file does exist then it will be replaced by the new file. If you already have some names and numbers in your file and wish to add to them then you should use the programme 'tel3' given later and NOT use this programme again.

Line 100 asks for a name and telephone number and takes input from the keyboard.

Line 110 checks for the name 'end' or 'END' which is used to indicate the end of the list. When found the channel is closed and the programme stops.

## 19

Lines 120 and 130 print the name and telephone number to the file using stream number four, e.g. *PRINT # 4;n\$* sends the contents of n\$ to stream number four followed by the *ENTER* character. Line 140 returns to line 100 and the cycle repeats again.

Now load and list the programme 'tel2' from the introductory section.

```
10 OPEN #4;"m";1;"telephone" IN
100 PRINT #4:
110 IF USR 432 = 0 THEN CLOSE # 4:STOP
120 INPUT #4;n$;t$
130 PRINT n$,t$
140 GOTO 100
```

Line 10 opens the file 'telephone' for input, This file input channel is 'linked' to stream number four. The *IN* part of the

*OPEN #* command indicates that the file is to be used for input. If the file cannot be found on the disc in drive number one then an error report is generated.

Line 100 sets the current stream to four ready for the call *USR 432*. This call returns the number of characters still to be read from the current stream.

Line 110 checks to see if the number of bytes not yet read from the file is zero. If it is, we have reached the end of the file and the channel is closed and the programme stops.

Line 120 inputs the next name and telephone number from the file (stream four).

Note that the items in the variable list of the *INPUT #* command (n\$ and t\$ in this example) must be separated by

semicolons and not commas. Alternatively each item can be input using a separate *INPUT #* command, for example, line 120 could be replaced by:

```
120 INPUT # 4;n$
125 INPUT # 4;t$
```

Note also that the *INPUT #* command requires that each item be followed by the *ENTER* character. When printing to a file this can be achieved by printing the items separately or by including an apostrophe as a separator for the items in the variable list, for example, lines 120 and 130 in programme 'tel1' could be replaced by:

```
120 PRINT # 4;n$t$
```

Line 130 prints the name and telephone number on the screen.

Line 140 returns to line 100 and the cycle repeats again.

Notice the similarity in the two programmes. The only difference is in the sources and destinations of the data

The list of telephone numbers can be extended by using the following programme:

```
10 OPEN #4;"m";1;"telephone" EXP
100 INPUT "Name? ";n$,"Tel. No.?" ;t$
110 IF n$="END" OR n$="end" THEN CLOSE # 4:STOP
120 PRINT #4;n$
130 PRINT #4;t$
140 GOTO 100
```

Save the programme on disc as 'tel3' and then run the programme. It works in the same ways as 'tel1' . Enter some extra

names and numbers and finish with 'end' or 'END' and any number. When it has been run you can display the updated file 'telephone' by using the programme 'tel2' as before.

This time line 10 opens the file 'telephone' on drive number one for expansion by using the *EXP* option.

When used with an "m" channel both the *OUT* and *EXP* options can be followed by a number. For the *OUT* option this number represents the maximum length of the new file. For example:

```
OPEN # 6;"m";1;"Carol" OUT 1000
```

will set up stream number six for output to a file 'Carol' on the disc in drive number one. The maximum length of the file is 1000 characters (bytes). If the number is omitted, or given as - 1, the disc system will allocate one half of the largest available space on the disc to the file. For the *EXP* option the number represents the length by which the file is to be extended. For example:

```
OPEN # 11;'m" ;2;"Laurie" EXP 256
```

will set up stream number eleven for output to the file 'Laurie' on the disc in drive number two. The output will be

added to the end of the file and will not be more than 256 bytes long. An error report will be generated if there is insufficient room on the disc for the expansion. If the length is omitted, or given as - 1 , the file is doubled in length or, if insufficient space is available, is expanded to fill the gap between the end of the file and the start of the next file on the disc.

## Random access files

The programmes just illustrated show the use of disc files for data storage. In these programmes the information is saved and retrieved one item after another. If you want to locate the third name and telephone number in the file you must first read and discard the first two names and telephone numbers. Altering the information is even more difficult. To improve this situation the Discovery disc system allows the use of random access files. In a random access file each set of data (called a record) is a fixed number of characters (which must be decided before you set up the file). You can then point to the set of data that you require. To create a random access file you use the *OPEN #* command with the *RND* option,for example:

```
OPEN # 10;"m";1;"Paul" RND 20,10
```

will create a random access file with a record length of twenty bytes and will hold 10 records i.e. the file will be two hundred bytes long. Unlike the *OUT* option, the

## 20

number of records must be given if you wish to create a random access file, if you do not wish to specify the number of records use - 1 , for example:

```
OPEN # 4;"m";1;"George" RND 64, -1
```

will create a random access file with a record length of sixty-four bytes. In this case the length of the file will be one half of the largest available space on the disc. The following programme creates a random access file and initializes it from the information in the data statements.

```
10 DIM f$(9)
20 OPEN # 4;"m";1;"random" RND 10,10
30 FOR i = 1 TO 10
40 READ f$
50 PRINT # 4;f$
60 NEXT i
70 CLOSE # 4
80 DATA "Dave", "Ian", "Laurie", "Rowland", "Eric"
90 DATA "Chris", "George", "Hector", "Paul", "Carol"
```

Note that the string is dimensioned to be nine bytes long, but when printed the *PRINT #* command adds an *ENTER*

character which makes the record length ten bytes. As there is an *ENTER* character at the end of each record the records can be input using *INPUT #* , so we do not have to read the record one byte at a time using *INKEY\$ #* .

To open an existing file for random access the *RND* option is used, with just the record length given, for example:

```
OPEN # 8;"m";1;"Dave" RND 128
```

assumes that the file exists and it is opened for random access with the specified record length, 128 bytes in the

example above. When open for random access you can read or write to the file at the current position of the file pointer. The *POINT #* command is then used to move the file pointer to a particular record in the file, for example:

```
POINT # 4;13
```

moves to the start of the thirteenth record of the random access file linked to stream number four. The following programme shows how records can be read from the file we have just created.

```
10 OPEN # 4;"m";1;"random" RND 10
20 INPUT "which record";n
30 IF n < 1 or n > 10 THEN PRINT "Record number out of range":GOTO 20
40 POINT #4;n
```

```

50 INPUT # 4;f$
60 PRINT "Record number";n;" is ";f$
70 INPUT "Another record (y/n)";r$
80 IF r$ = "y" THEN GOTO 20
90 CLOSE # 4

```

Now run the programme and enter some record numbers. The corresponding names should be displayed.

Line 10 opens the file 'random' for random access with a record length of 10. The string, f\$, is not dimensioned this time as this programme only reads from the file. If the programme was also writing to the file then f\$ would have to be

dimensioned as this makes sure that it is always the correct length.

Line 40 moves the file pointer to the start of the required record.

Line 50 inputs the record.

Note that the record numbers do not have to be in order and this makes random access files very powerful and well worth the effort necessary to learn how to use them.

## The erase command

This command allows unwanted files to be removed from the disc. For example the command:

```
ERASE 1;"concorde"
```

removes the file 'concorde' from the disc in drive number one. If this short form of the command is used, and the file does not exist then the error message 'file not found' will be returned.

If the 'm' channel is specifically included e.g.

```
ERASE "m";1;"concorde"
```

and the file 'concorde' does not exist, then no error is produced. This is to provide compatibility with Sinclair microdrives.

## REFERENCE GUIDE

As you will have found, if you have worked through the manual's introductory sections, the Discovery system provides a disc operating unit which is not only powerful and flexible but also easy to use. In this section we describe the commands together with all the options available with each command. Many of the commands require similar expressions to be included, for instance whenever a drive number is required it can be 1,2,3,4 or 5. So when you see <drive> in a command replace it with 1,2, 3, 4, or 5 according to the drive number which you wish to use.

The use of this 'notation' will make it easier for you to understand the use of the commands.

### 1) Notation

1) <drive>

The Discovery unit can support two disc drives which are normally numbered 1 and 2. Drive 1 is normally on the left hand side. Drive 3 can also be used to represent Drive 1 and Drive 4 to represent Drive 2. This is usually used for copying files or discs on a single drive system. Drive 5 refers to the RAM disc (see later).

## II) <stream>

The Spectrum uses streams to transfer data from a source to a destination. Every stream used must be numbered between 0 and 15 (an integer). Stream numbers 0, 1 and 2 are already reserved for use by the Spectrum system.

Stream number 3 is set up by the Spectrum for use with the ZX printer. However this may be changed to use the

Discovery printer port. See *OPEN #* command. Normally streams 4 to 15 are available to the Discovery user.

## III) <filename>

Every file on a Discovery disc has a name. The name is made up of a string expression of between 1 and 10

characters e.g.: "utility" or "READ ME".

NOTE: "quotes" is different from "QUOTES", upper and lower case letters are not taken to be identical.

## IV) <channel spec.>

The Spectrum computer uses a number of channels to represent 'hardware'. For instance "K" represents the keyboard.

The Discovery unit adds a number of additional channels to the Spectrum. The channel specifications available are:

"b" - input and output to the parallel port, characters are sent and received as 8 bits with no translation.

"CAT" - this channel gives access to the disc catalogue file. The complete channel specification is "CAT" ;<drive>.

Care must be taken if writing to the catalogue file as if this is corrupted then files on the disc may be lost. This channel is best opened for random access with a record length of 16, when *POINT #* can be used to move to a given entry.

"CODE" - this channel writes or reads directly from memory. Use *POINT* to select the memory location at which you wish to read or write. This channel provides a method of reading the shadow ROM and also gives access to the disc system hardware.

"d" - this channel treats the disc as a single entity and is used by the *MOVE* command. The full specification is "d";<drive> .

"j" - this channel turns the joystick port on and off and is only used by the *FORMAT* command.

"K" - the keyboard (also outputs to the lower part of the screen).

"m" - disc file. The complete channel specification is given below as <file spec>.

"P" - the printer

"S" - the top part of the screen.

"t" - input and output to the parallel port. For input the top bit is set to zero giving codes between 0 and 127. For output character codes less than 32 are ignored (except for CR which causes a CR or a CL/LF pair to be sent, see below).

Character codes greater than 127 are expanded if they represent BASIC tokens and are printed as '?' if they represent graphics characters. The full specification is "t";<status> .

## 22

If status is 0 then when a CR is encountered a CR/LF pair is sent, if It is 2 then only a CR is sent. If it is 1

then the ZX printer is emulated. The status can be omitted and in this case the default is 0.

"#" - this 'stream ' channel allows you to open one stream to another. The full specification is "#";<stream>. For example,

```
OPEN # 4 ;"#" ; 1
```

'links' stream four to stream one. For this channel the quotes and the separator can be omitted and it can be specified as # <stream:> .

NOTE A) Upper or lower case letters can be used in the channel specification and so, for example, "k" and "K" both refer to the keyboard channel.

NOTE B) After using a channel other than the screen the screen attribute commands such as - *INK* and *PAPER* - may not work. To avoid this problem include the command *PRINT* ; before using them.

V) <file spec.>.

In order to precisely stipulate a given file on the Discovery system, the file name alone is insufficient. You must also provide the drive number and inform the system that the file is on the disc unit. The complete file specification is as follows:

```
"m";<drive>;<filename> .
```

The "m"; can be omitted from the specification as "m" is the default channel.

## 2) Keywords used in the Discovery system

*CAT* <drive>

This command displays the catalogue of the disc in the specified drive. The number given at the end of the list of files

is the space still available on the disc in kilobytes (rounded down). e.g.:

```
CAT 1
```

displays the catalogue of the disc in drive number 1 on the screen provided that stream 2 is allocated to the screen.

*CAT # <stream>;<drive>*

This command sends the catalogue of the disc in the specified drive to the given stream. The stream must have been previously opened for output e.g.:

```
OPEN #4;1;"directory"  
CAT# 4;2  
CLOSE # 4
```

This will put the catalogue of drive 2 onto a file called "directory" on drive 1 .  
*CAT <drive>* is an abbreviation for *CAT # 2; <drive>* and so if stream 2 is reallocated the output will be sent to the 'new' channel.

*CLEAR #* or  
*CLEAR # <stream>*

*CLEAR #* clears out open streams and makes them available for further use.

*CLEAR # <stream>* has the same effect, but, only on the specified stream.

*CLOSE # <stream>*

This command closes the specified stream and frees it for further use. In the case of disc files it is important to close a file when access is finished, particularly when writing to a file. Closing a file writes any information still in the buffer to the disc file. For serial files it also updates the catalogue entry setting the file length to the current position. Failure to close a file will mean that information is lost and may also mean that the catalogue entry is incorrect.

Note that when you close streams 0, 1 , 2 or 3, they are closed and then automatically reopened to their default channels.

*CLS #*

This command clears the screen and sets the display attributes to their defaults:

## 23

```
PAPER 7  
INK 0  
BORDER 7  
FLASH 0  
BRIGHT 0  
OVER 0  
INVERSE 0
```

*CODE <start address>,<no.of bytes>*

This is an extension to the *LOAD \**, *VERIFY \** and *SAVE \** commands which allows a block of memory to be saved, loaded or verified. When saving a block of memory the start address and number of bytes must be given in decimal. When loading or verifying, the start address and number of bytes may be omitted,

although you cannot omit the start address if you wish to specify the number of bytes. Any value that is omitted is obtained from the file and will therefore be the same as when the block of memory was saved. Loading a file overwrites the memory until all of the file is loaded and may overwrite information that is needed. If you wish to load a file into a block of memory whose size is fixed, this size should be given in the command. The file is then only loaded if enough room is available. For example:

```
SAVE *1;"bytes" CODE 1234,50
```

Saves 50 bytes starting at 1234 to the file called "bytes" on Drive 1.

*DATA* <array name> ( )

This is an extension to the *LOAD* \*,*VERIFY* \* and *SAVE* \* commands which can be used to save and load an array.

Numeric and string arrays can be saved and reloaded. Note that the ( ) must be included. e.g.:

```
SAVE * 1 ;"fred" DATA A( )
```

saves the numeric array A to a file called 'fred' on Drive 1 .

*ERASE* <File spec>

This command erases the file specified.

The filename is removed from the directory and the space on the disc occupied by the file is made available for

further use e.g.:

```
ERASE 1 ; "Rowland"
```

erases the file called "Rowland" from disc 1 and generates an error report if the file is not found.

The full form of the

command - *ERASE* "m";1;"Rowland" - produces no error if the file is not found.

*FORMAT*

This command is used to set up channels "j" , "m" and "d" .

*FORMAT* "j";<status>

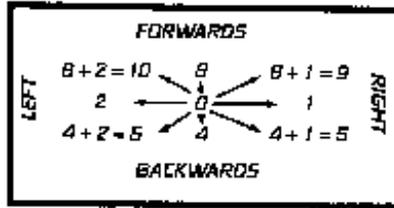
Is used to turn the joystick on and off. <status> = 0 turns the joystick off, while <status> = 1 turns the joystick on. e.g.:

```
FORMAT "j" ; 1
```

turns the joystick on. When the joystick has been turned on the current position of the joystick and the state of the fire button can be obtained by using *IN* 31 . For example if using:

```
1000 LET stick= IN 31
```

the values of the 'stick' can be interpreted from the diagram



Pressing the fire button adds 16 to the values of 'stick' shown above.

*FORMAT "m";<drive>;"<disc title>"*

Is used to format a disc. If the disc is already formatted then you will be asked to confirm that you wish to 'destroy' the disc.

*FORMAT 1;"Rodger"*

will format the disc in drive 1 and give it the title "Rodger".

NOTE: ' m ' may be omitted since it is the default channel.

## 24

*INKEY\$ # <stream>*

This command reads one character from the specified stream. The stream must have been previously opened for input. If the stream is associated with an "m" channel the command will only return a null string when the end of file has been reached. Other 'slow' channels will return a null string when a character is not available. Slow channels are "K", "t" and "b".

*INPUT # <stream>;variable 1;variable 2;....*

This command inputs values from the specified stream into the variables given.

The command is identical to the BASIC *INPUT* command and may therefore also try to output to the specified stream. This is possible provided that the stream has been opened for output as well. Care should be taken if using this facility and it is recommended that it is not used when inputting from a file. In this case items in the variable list must be separated by semicolons and not commas, to make sure that there is no output to the stream. In the file the values to be read must be separated by the *ENTER* character and care must be taken to ensure that this is done when writing the file. See *PRINT* below. The stream being used for input can be changed from within the command, for example:

*INPUT #5;A; # 7;b\$c*

will get the value of A from stream number 5 and the values of b\$ and c from stream number 7.

*LINE <line number>*

This extension to the *SAVE \** command can only be used when saving a programme. It makes the programme run from the specified line number when it is reloaded e.g.:

*SAVE \*1;"Chris" LINE 10*

saves the current programme onto the disc in drive 1 as a file called "Chris" so that when it is subsequently loaded it will automatically run from line 10.

### *LOAD* \* <channel spec.>

This command loads a programme and variables from the channel specified. It can also be used with a number of extensions, see *CODE*, *DATA* and *SCREEN\$*.

### *LPRINT* # <stream>;variable 1,variable 2,....

This command prints the values of the variables in the variable list to the specified stream. It works in exactly the same way as the *PRINT* # command.

### *LPRINT*

*LPRINT* works in the same way as *LPRINT* # <stream> above but always sends the variables to stream 3.

### *LLIST*

This lists the programme to stream 3 which is set up by the Spectrum to be the ZX printer. To list programmes using the Discovery printer port, stream 3 must be opened to the 't' channel e.g.:

```
OPEN #3;"t"
```

```
LLIST
```

### *MERGE* \* <channel spec.>

This command merges a programme and variables from the specified channel with the programme and variables already in the memory. Any programme lines or variables that are duplicated will be overwritten by the new ones.

### *MOVE A TO B*

This command transfers information from an input channel or stream ,A, to an output channel or stream ,B. The process continues until end of file occurs in the input channel. For input from a 'slow' channel like the keyboard or the parallel port, an end of file must be generated and this is done by pressing *SHIFT* and *ENTER* simultaneously. File transfers using *MOVE* will be faster and involve fewer disc swaps if the Spectrum is first reset using *PRINT USR 0*. This enables the *MOVE* command to use the largest buffer possible. If you have a programme in memory and a Spectrum reset is not convenient, some space can be freed by using the *CLEAR* command.

```
OPEN # <stream>;<channel spec.> <access>
```

This command associates the specified stream with the channel given. Possible types of access are:

*IN* used for input only.

*OUT* used for output only.

For the "m" channel the following extra options are available.

*EXP* - used to expand a serial file. This option is output only and automatically adds information to the end of the file. If the file does not exist then an error report will be generated.

*RND* - used for random access files and allows input and output.

The type of access can be omitted and in this cassette defaults for each of the channels are:

"K" - *IN* and *OUT*

"S" - *OUT*

"P" - *OUT*

These channels always have these access options, even if you try to change them. You could open a stream for input from the "K" channel, for example, by using

```
OPEN #4;"#";1 IN
```

which links stream four to stream one for input only.

The defaults for the other channels are:

"b" - *IN* and *OUT*

"CAT" - *IN*

"CODE" - *IN* and *OUT*

"t" - *IN* and *OUT*

"#" - *IN* and *OUT*

"m" - the "m" channel behaves in the same way as the "m" channel on the microdrive system. If the file already exists it is opened for input, while if it does not exist the file is created and the channel is set up for output to the file. This is not the same as the *OUT* option where if the file exists it is overwritten.

For the "m" channel with the *OUT* option a length can be specified. This length is the maximum size of the file and is used in the initial catalogue entry. If unspecified (or given as - 1 ) the length is set to one half of the largest space remaining on the disc. While if given as 0 all of the largest space remaining will be allocated to the file. When the channel inclosed the file length is correctly written. In a similar way the *EXP* option can have a length specified. This time the length given is the maximum size of the expansion required. If there is insufficient room for the expansion then an error report will be generated. If a length is not specified (or given as - 1 ) the file will be doubled in length or if insufficient space is available It will be expanded to fill the gap between the current end of the file and the start of the next file on the disc.

The *RND* option is used in two ways, to create a random access file and to open an existing random access file. To create a random access file two parameters must be given. These are the record length and the number of records required in the file. The number of records can be given as 0 or - 1 , when it behaves in the same way as for the *OUT* option. For example:

```
OPEN # 4;"m";1;"Dave" RND 32,30
```

creates a random access file 'Dave' with a record length of 32 bytes and with room for thirty records (the length of the file will be  $30 * 32 = 960$  bytes). To open an existing random access file only the record length is given. For example, to open the file ' Dave' created above use:

*OPEN #4;"m";1;"Dave" RND 32*

For the "t" channel a printer width can be specified by using the RND option, for example:

*OPEN # 4;"t";0 RND 80*

sets up stream four for output to the printer and sets the printer width to eighty characters. If no printer width is specified then It defaults to 0 which means that no check on line length is carried out. This option only applies to output using the "t" channel, It has no effect on input.

## 26

*POINT # <stream>;<record number>*

This command is used with random access files to set the file pointer to the start of a given record.

Records are

numbered consecutively starting from 1.It can be used with serial files provided that they have been opened as random

access files with a record length of 1. In this case <record number> is used to point to a given byte in the file.

*PRINT # <stream>;variable1; variable 2,....*

This command prints the values in the variable list to the specified stream. The items in the variable list can be separated by any of the separators used with the *PRINT* command .When writing to a file which is to be read using another programme, the items in the file must be separated by an *ENTER* character and they must therefore be printed separately or separated by apostrophes in the print list. Printing to different streams can be carried out from the same print statement e .g . :

*PRINT # 5;"SPECTRUM"; # 8;"disc" ' # 2;"system"*

will send "SPECTRUM" to stream 5,'disc + ENTER" to stream 8 and 'system + ENTER' to the screen (stream 2).

*SAVE \* <channel spec.>*

This command saves the current programme and variables to the channel specified. It can also be used with a

number of extensions,see *CODE*, *DATA*, *LINE* and *SCREEN\$*. If a disc file is specified and it already exists on the disc it is replaced. The 'old' disc file is not overwritten and so it is not immediately lost, e.g.:

*SAVE \*1 ;"John" CODE 16384,6912*

saves the current screen to a file called "John" on drive 1 .

Note that a *CLEAR* command before saving will delete all variables and allow just the programme to be saved.

*SCREEN\$*

This extension to the *LOAD \**, *SAVE \** and *VERIFY \** commands allows the screen to be saved and reloaded. It is equivalent to *CODE 16384,6912* e.g.:

*LOAD \*1;"John" SCREEN\$*

loads the previously saved file "John" from drive 1 to the screen.

*VERIFY \* <channel spec.>*

This command checks that the information that has been saved is the same as the information in the computer's memory and gives an error message if it is not. It can be used with a number of extensions, see *CODE* , *DATA* and *SCREEN\$*

### 3) The autorun Facility

After turning on the Spectrum, or after resetting the machine by typing:

*NEW ENTER*

it is possible to automatically run a programme from the disc in drive number 1 . This is done by typing:

*RUN ENTER*

when the disc system will look for a file called 'run' on the disc in drive 1 . If it is found it will be loaded and should run automatically The required programme must have been saved on disc with the filename "run" with the *LINE* extension used to indicate the starting line of the programme. This is done with the command:

*SAVE \*"m";1;"run" LINE < line number> ENTER*

If the *LINE* extension is not used the file 'run' will be loaded automatically, but it will not be run.

### 4) Useful *USR* routines

*USR 0* - this resets the Spectrum, but not the disc system.

## 27

*USR 14070*- this resets the disc system.

*USR 8*- this returns the version number of the disc software.

*USR 432*- this returns the number of bytes remaining in the current stream. If the stream is of unknown length then it returns the value - 1 .

*PRINT #4; < .....>* for example. leaves four as the current stream.

NOTE 1) *INKEY\$ # <stream>* leaves the current stream unchanged.

NOTE 2) The commands *PRINT #4;PRINT USR 432* will not work as the second *PRINT* resets the current stream to the screen. To do this use:

```
PRINT #4;
LET length=USR 432:PRINT length
```

This can be more usefully used to test for the end of a file without inputting a character using *INKEY\$*, for example:

```
PRINT #4;
IF USR 432 = 0 THEN PRINT "eof"
```

## 5) Compacting a disc

The disc system stores the files on the disc on consecutive sectors. After some use, saving and erasing files, the space on the disc becomes fragmented with small numbers of sectors that cannot be used. To make use of these sectors the disc must be compacted. This process moves the files so that they are adjacent and therefore combines all of the free space. To compact a disc in drive 1 type:

```
MOVE "d";1 TO "d";1 ENTER
```

WARNING: IF THE POWER IS DISTURBED OR THE SPECTRUM CRASHES DURING THE COMPACTING OF A DISC THE FILES ON THE DISC MAY BE DESTROYED IT IS WISE TO MAKE COPIES OF IMPORTANT FILES BEFORE COMMENCING THE COMPACTING PROCEDURE.

## 6) The catalogue file

The catalogue file can be accessed using the "CAT" channel. The file consists of a series of sixteen byte entries, one for each file on the disc, and an end marker. The first entry always refers to the catalogue file. Each sixteen byte entry consists of three two-byte numbers followed by a string of 10 bytes. First number-the low twelve bits of the number represent the number of bytes in the last block of the file - 1. The top four bits are used by the system. Second number - the start block of the file. Blocks on the disc are numbered from 0, not 1. Third number-the end block of the file (inclusive). The string is the filename, or in the case of the catalogue file, the disc title. The end marker is another sixteen byte entry but only the second and third numbers (byte numbers 3 to 6 inclusive) are significant. The second number is the number of blocks on the disc and the third number is 65536 (in hex FFFF.) The third number can be used to detect the end of the catalogue file. When a disc is formatted the catalogue file is set up for 62 files. If you obtain a disc full error message, but the CAT command shows that there is still space available, then the catalogue file may be full. It can be extended by using the commands:

```
OPEN # 4;"CAT" EXP < length> ENTER
CLEAR # 4 ENTER
```

when, if possible, the catalogue file will be extended. The length can be calculated from the formula  $16 * \text{No. of extra files}$ . If no length is specified the catalogue file is doubled in length or if insufficient room is available is extended as far as possible..

Note - It may also help to compact the disc as shown earlier.

## 7) The RAM disc

The Discovery disc system allows you to set aside part of the Spectrum's memory as a RAMdisc. To do this type:

```
CLEAR 32767 ENTER  
FORMAT 5;"ramd"
```

The *CLEAR* command reserves the memory needed for the RAMdisc. The disc system recognises drive number five as the RAMdisc and formats it accordingly. Use the following programme to transfer some files from the 'First disc' to the RAMdisc.

28

```
10 READ f$  
20 IF f$ = "end" THEN STOP  
30 MOVE "m";1;f$ TO "m";5;f$  
40 GOTO 10  
50 DATA "tel1","tel2","tel3","telephone","joysketchb","end"
```

Now load the programme "tel2" from the RAMdisc using:

```
LOAD *"m";5;"tel2" ENTER
```

and then make the following change:

```
10 OPEN # 4,"m";5;"telephone" IN
```

and run the programme. You will notice how much faster the operations are when using the RAMdisc. This is particularly useful when file processing, providing of course that the file will fit into the available memory. The file to be used should be moved to the RAMdisc, manipulated from there and then moved back onto the disc when the processing is completed.

## Appendix 1

### Transferring programmes from tape to disc

The most straightforward method of transferring from tape to disc is to load the programme from tape and then save it to disc in the usual way. This will not work however with programmes that run automatically and in these cases you can try merging the programme. Suppose that you wish to transfer a programme called 'MAGNUM'. First type:

```
NEW ENTER
```

To remove any unwanted programme or variables, and then type:

```
MERGE "MAGNUM" ENTER
```

If successful you can then save the programme to disc in the usual way. If this fails you will need to use one of the programmes commercially available to carry out the transfer. Some programmes on cassette come

with instructions for transfer to microdrive and the same instructions will usually allow you to transfer the programme to disc.

## Appendix 2

### Error reports on the Discovery system

The following error reports, with their error codes are produced by the Discovery disc system.

a- Invalid device name-the channel identifier is incorrect. Allowed device names are listed in the reference section.

b- stream already open - the stream specified is already in use

c- invalid drive number-the drive number given is incorrect. Allowed drive numbers are 1 , 2, 3, 4 or 5.

e - write protected - the disc in the

f- disc full -there is insufficient space on the disc. Try compacting the disc.

g-disc I/O error-this error occurs when there is a problem reading from or writing to the disc. Possibilities are: disc not in the drive, specified drive not connected or corrupted information on the disc.

h -file not found -the filename given has not been found on the disc in the specified drive.

i- hook code error- this occurs when using machine code with microdrive hook codes. It should not occur when using BASIC.

j- file size error - this occurs if you try to verify, or write past the end of a file. When a file is opened a length is always set even if it is half of the disc, see *OPEN #* in the reference section.

l- verification failed.

m - wrong file type - the file specified in the command is not the correct type.

■ - ram corrupt- this message occurs if there is an internal inconsistency e.g. if a channel contains the wrong channel length.

?- no message - in this case the error code is given: There is no message define for this error code.

## Appendix 3

### Making connections to the Discovery system

There are a number of different input/output ports on the Discovery disc unit and these are now briefly described.

### Peripheral through connector

The Spectrum expansion connector is carried through to the right hand side of the Discovery unit and is labelled .

'Peripheral through connector'. All lines are available except A14 (The A side of the connector is the component side of the board and A14 Is the NMI line). When held high the ROMCS line, B 25, disables the Spectrum ROM-and the Discovery shadow ROM.

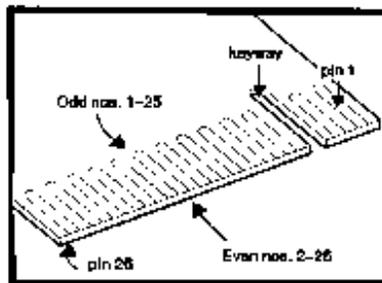
### Joystick connector

The joystick connector is positioned towards the rear on the right hand side of the Discovery unit and can be used with any Atari style joystick.

### Parallel printer port

The edge connector for the Parallel Printer Port Is also positioned on the right hand side of the unit. It needs a standard 26 pin IDC connector and the pin connections are:

- 1 - strobe
- 3- D0
- 5-DI
- 7- D2
- 9- D3
- 11 - D4
- 13- D5
- 15-D6
- 17-D7
- 19-Acknowledge
- 21 - Ready
- odd numbers 21 to 25 and 26- no connection
- Even number 2-24-0V



To use a parallel printer you will need a printer cable. A cable with an edge connector at one end and a Centronics printer connector at the other can be obtained from Opus Supplies Ltd., its part number is PCDC1 . To connect two Spectrum computers you need a cable with an edge connector at each end. The connection is straight through pin to pin and so the edge connectors can be crimped directly onto each end of the cable.

The Discovery range from Opus Supplies has been specially developed with expansion in mind. 'Discovery 1' can be upgraded to a dual-drive system by purchasing the 'Discovery +'. This upgrade package will be fitted by Opus Supplies. Under no circumstances should a purchaser attempt to fit this upgrade system as this will invalidate the guarantee. 'Discovery 2' is available as a purpose built dual-drive system.

## Guarantee

This Opus product is guaranteed for a period of 12 months from the date of purchase, subject to the conditions listed below:-

1. The guarantee shall only be effective if the whole guarantee form is completed and returned to Opus Supplies Limited within ten days of the product's purchase.
2. The guarantee applies only to the original purchaser and is not transferable. The warranty applies only to products purchased from an authorized dealer to whom the equipment was sold by the company or direct from Opus Supplies.
3. All repairs and modifications to the unit must be undertaken by the company via the dealer network. The warranty is immediately invalidated if this procedure is not followed. Where possible claims under the guarantee should be referred to the dealer from where the unit was purchased. If this is not possible you should contact Opus direct. When making a claim the customer's portion of the certified guarantee must be produced.
4. Customers are liable for the carriage costs of units returned under the claims procedure
5. If, after testing by the company, no fault can be found with the system and the unit complies with the published specification, the company reserves the right to charge a reasonable fee for testing and returning the unit.
6. The liability of the company under this guarantee is limited to the cost of repair or complete replacement (at the discretion of the company) of the defective equipment.
7. The company does not accept responsibility for any loss or damage during transit to or from the dealer or the company, but in this event every effort will be made to investigate complaints of loss or damage if they should arise.
8. This guarantee does not affect the purchaser's statutory rights.

## ADDITIONAL INFORMATION

### THE *MOVE* COMMAND

On page eleven of the operating manual - the MOVE command - an amendment has been made to the instructions for transferring data from the "K" channel to the "S" channel and to the "K" channel. All slow

channels are now buffered. It will therefore be necessary to press *CAPS SHIFT* and *ENTER* together before the characters you have entered at the keyboard will appear on the screen

## THE "CAT" CHANNEL

When opening a stream to the "CAT" channel, *POINT #* should be used to select the required record before reading or writing data.

## VIDEO MONITOR OUTPUT

The video monitor output positioned at the rear of your Discovery unit provides a monochrome composite video signal and should therefore only be used with a video monitor. If using a normal television receiver you should connect the aerial lead supplied with your computer into the TV socket provided at the rear of your Spectrum as outlined in the assembly instructions.

## DEFAULT CATALOGUE

the default catalogue size has been extended from 62 to 110 files.

## DRIVE NUMBERS

Drive number 4 has not been implemented. There is no requirement for this and you should use drive numbers 1 and 3 for copying on single drive systems.

## INDEX

Assembly 3, 4  
Autorun 6, 15, 27  
  
BREAK (SHIFT-SPACE) 10, 18  
  
catalogue file 11, 23, 28  
CAT 5, 10, 11, 23, 28  
CAT # 23  
channels 11, 12, 16, 17, 18, 22  
    "b" 12, 15, 16, 22, 25, 26  
    "CAT" 22, 26, 28  
    "CODE" 22, 26  
    "d" 9, 22, 28  
    "j" 15, 22, 24  
    "k" 11, 16, 18, 22, 25, 26  
    "m" 11, 12, 16, 22, 23, 25, 26  
    "p" 16, 22, 26  
    "S" 11, 12, 16, 18, 22, 26  
    "t" 12, 16, 22, 25, 26  
    "# " 23, 26  
binary channel 16  
default access options 26  
slow channels 25  
text channel 16

CLEAR 28  
CLEAR # 19,23,28  
CLOSE # 8,9,18,19,20,21,23  
CLS # 7,13,23  
CODE 13,14,24,25,27  
compacting a disc 28  
connections 30  
Copying a disc 8  
Copying a file 11,12

DATA 13,14,24,25,27  
Disc care 8  
Disc drive number 5,11,12,16,22  
Disc title 5,11,24

End of file 10,12,13,19,25,26  
ERASE 21,24  
Error Reports 6, 12, 13, 19, 20, 26, 29  
Expanding a file 12, 20  
EXP 20, 26, 28

Filename 22  
File pointer 21 , 27  
File specification 23  
File transfer 12, 15  
FORMAT 5, 7, 11 , 15, 24, 28

IN 8, 19, 26  
INKEYS# 10, 18, 21, 25, 28  
INPUT# 8, 19, 20, 21 , 25

Joystick 7, 15, 16, 30

LINE 6, 10, 15, 25, 27  
LLIST 9, 17, 25  
LOAD \* 6, 7, 13, 14, 15, 24, 25, 27  
LPRINT 9, 17, 25  
LPRINT # 25

MERGE\* 10, 14, 15, 25  
Microdrive 9, 26  
MOVE 9, 11, 12, 25, 26, 28, 29

OPEN# 8, 9, 18, 19, 20, 21, 26  
OUT 8, 19, 20, 26

Parallel port 30  
POINT# 21, 22, 27  
Printer 9, 12, 30  
Printer width 26  
PRINT# 8, 9, 18, 19, 20, 21, 27, 28

RAM disc 22, 28, 29  
Random accessible 8, 20, 21, 27  
RND 20, 21, 26  
  
SAVE\* 5, 6, 7, 8, 9, 13, 14, 15, 24, 25,27  
SCREEN\$ (SCRN\$) 7, 13, 14, 25, 27  
Sector 10, 11  
Streams 16, 17, 18, 19, 22, 23, 26, 28  
  
Tape to disc 6, 7, 10, 29  
Track 10  
  
USR routines 8, 19, 27, 28  
  
VERIFY\* 7, 13, 14, 15, 24, 27