sinclair
ZX SPECTRUM
Microdrive and
Interface 1 manual

sinclair

ZX INTERFACE 1

and

ZX MICRODRIVE

Designed and prepared by
Cambridge Communication Limited

Note
The names **ZX Microdrive** and **ZX Interface 1** are trademarks of
Sinclair Research Limited
Stanhope Road
Camberley
Surrey
GU15 3PS

Tel: Camberley (0276) 685311

First Edition 1983
© Sinclair Research Limited
Front cover illustration by John Harris of Young Artists

THE ZX INTERFACE 1 . . .

incorporates the three functions of Microdrive controller, local area network and RS232 Interface. Connect it to your Spectrum and you can control up to eight Microdrives, communicate with other computers and drive a wide range of printers.

THE ZX MICRODRIVE . . .

gives you fast access to a large memory. Each Microdrive can hold up to 100K bytes using a single interchangeable cartridge.

THIS MANUAL . . .

introduces the idea of using BASIC as an *operating system* as well as a *programming language.* This way, BASIC can be used to set up communication links, to send and receive data along them and to manipulate files. This adds enormously to its power to perform tasks in which:

- data is stored and retrieved,
- data and programs are sent to peripherals,
- communication links are established with other Spectrum computers or with other types of computer.

The combined use of BASIC as an operating system and a programming language gives it a power and flexibility, and also an ease of use, to be found in few larger computers.

You should read the chapters on the Microdrive even if you have not bought one, since they introduce various concepts (about *channels* and *streams* and their use) which you will need to understand.

This manual should only be read *after* you have familiarised yourself with the ZX Spectrum Introduction booklet and with chapters 1 to 22 of the BASIC programming manual.

Contents

Chapter 1	**Setting up your ZX Interface 1**	**5**
Chapter 2	**Setting up your Microdrive**	**9**
	handling cartridges	
	protecting programs and data stored in cartridges	
	the cartridge lifespan	
Chapter 3	**Starting with your Microdrive**	**15**
	auto-run	
	the catalogue	
	loading programs	
Chapter 4	**Programs and the Microdrive**	**17**
	saving, verifying, loading and merging programs	
	erasing programs	
	formatting and naming blank cartridges	
	setting up your own auto-run facility	
Chapter 5	**Data, channels and streams**	**21**
Chapter 6	**Data and the Microdrive**	**23**
	opening and naming a data file	
	entering data	
	closing a file	
	reading back from a file	
	notes on **PRINT** and **INPUT** — *separators*	
	— *changing streams*	
	— *setting colours*	
	reading the file catalogue	
	protecting a file	
	extending a file	
Chapter 7	**The Local Area Network**	**29**
	setting up a network	
	programs and the network	
	data and the network	
	broadcasting	

Chapter 8	**Using the RS232 Interface**	**35**
	connecting peripherals to the RS232 Interface	
	t and b channels	
	the t channel	
	the b channel	
	sending control codes	
Chapter 9	**The MOVE statement**	**39**
	the printer server program	
Appendix 1	**The net game**	**41**
Appendix 2	**System variables**	**45**
Appendix 3	**Microdrive and network channels**	**47**
Appendix 4	**RS232 connections**	**49**
Appendix 5	***Reports***	**51**
Appendix 6	**The extended BASIC**	**55**
Index		**59**

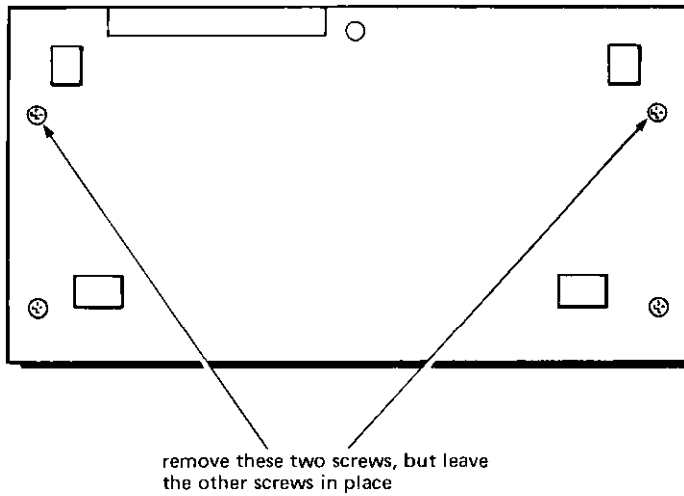
Setting up your ZX Interface 1

Unpacking the ZX Interface 1 you will have found:

- this booklet,
- the Interface itself (with two captive screws on the underside),
- a ribbon cable about 8cms long (for connecting the Interface to a Microdrive),
- a lead with a jack plug at either end (for setting up a network).

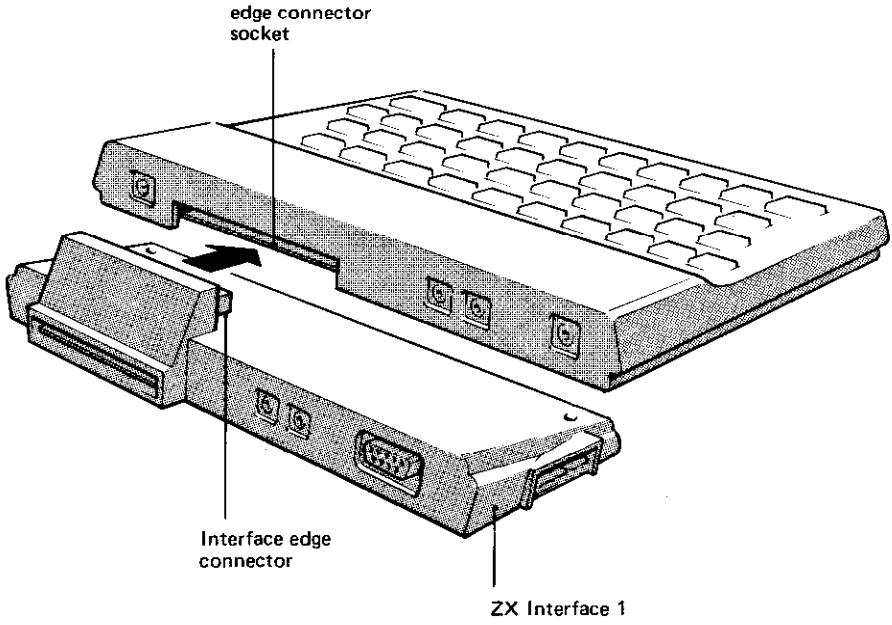
To attach the Interface to your computer you should:

1. Unplug the computer.
2. Disconnect the computer from its peripherals.
3. Using a Posidrive screwdriver remove the two screws on the underside of the computer, as shown in the diagram below. (You will only need these screws again if you later disconnect the Interface from the computer.)

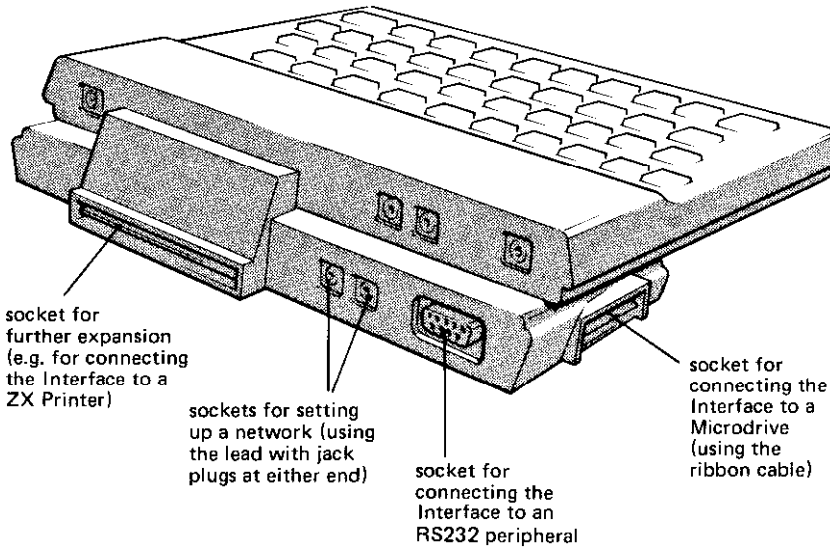


4. Push the Interface edge connector firmly into the socket at the back of the computer.
5. Screw the Interface to the underside of the computer using the two captive screws supplied.

You can now reconnect the computer to its peripherals, and plug it in.



The diagram below shows what the various sockets on the Interface are for.



You can now either:

– set up a Microdrive see page 9,

or

– set up a network see page 29,

or

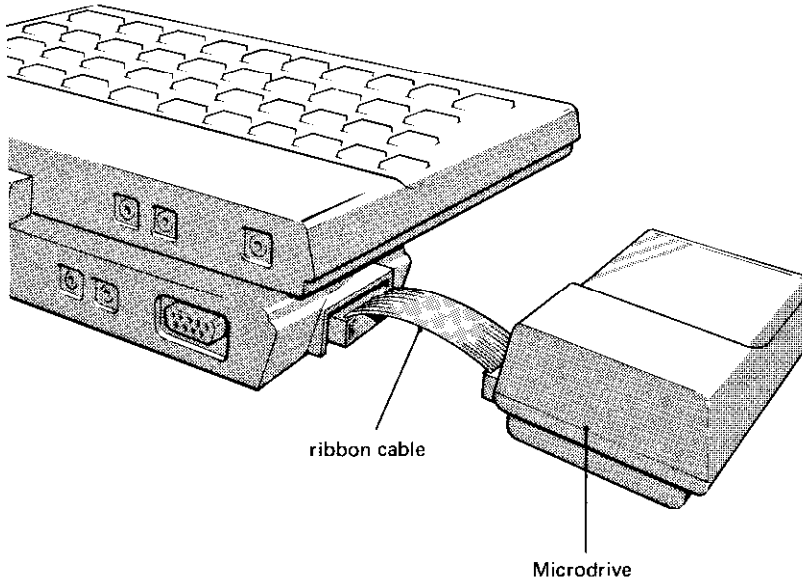
– connect the Interface to an RS232 peripheral see page 35.

Setting up your Microdrive

Unpacking the Microdrive you will have found:

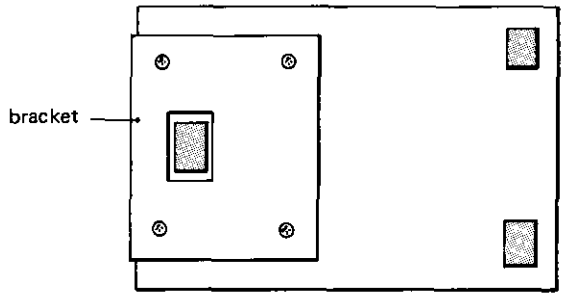
- the Microdrive itself,
- an electrical connector (for connecting this Microdrive to another one),
- a pre-recorded demonstration cartridge.

To attach the Microdrive to the Interface, push one end of the ribbon cable (supplied with the Interface) firmly into the socket on the side of the Interface. Then push the other end of the cable into the socket in the side of the Microdrive.

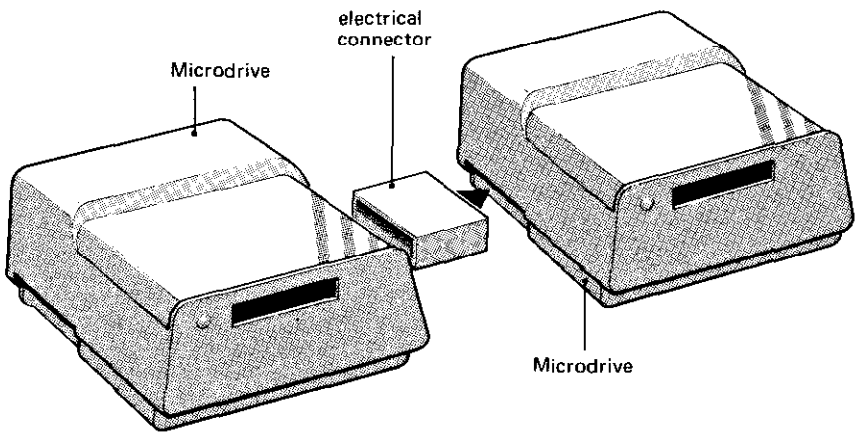


If you have several Microdrives, the second one can be connected to the first as follows.

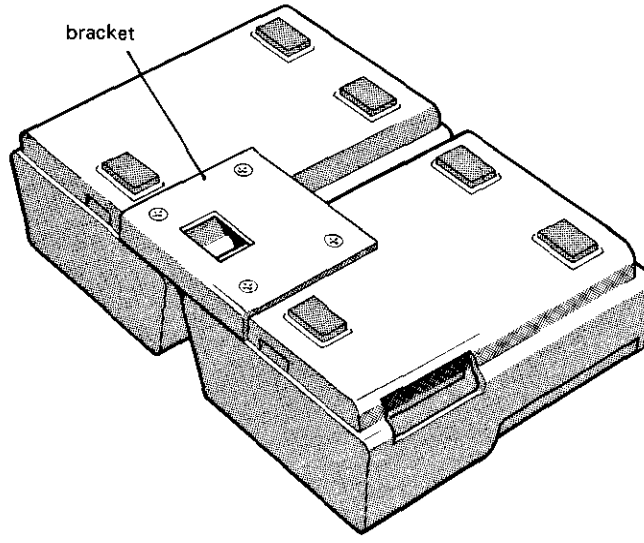
1. Disconnect the Microdrive from the computer.
2. Turn both Microdrives upside down. You will find that each has a bracket screwed to the underside.



3. Unscrew and remove both brackets.
4. Turn the Microdrives the right way up again.
5. Push one of the electrical connectors firmly into the socket on the side of one of your Microdrives (as shown below).
6. Push the second Microdrive up against the first, so that the other end of the electrical connector pushes into the socket on the side of the second Microdrive.



7. Carefully turning the Microdrives over again, screw back one of the two brackets – but this time so that it *links up* the two Microdrives. (The remaining electrical connector, bracket and screws can now be kept for linking up a further Microdrive.)



8. Turn the Microdrives the right way up again, and reconnect the one on the right-hand end to the computer.

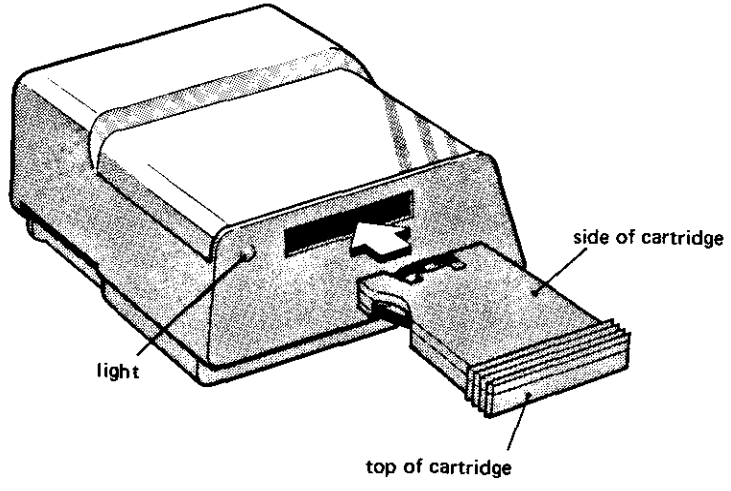
Note that the Microdrive nearest the computer is always known as Microdrive 1, and the next along is Microdrive 2, and so on.

Notice also that on the front of each Microdrive there is a light. This will come on whenever the Microdrive is running.

Handling cartridges

Every cartridge comes in a protective box; and should always be kept in its box when not in use.

When you take a cartridge out of its box (being careful not to touch the tape itself) you will see that it has a label on the top and another on the side.



Push the cartridge firmly into the slot in the front of the Microdrive, making sure that the label on the side faces upwards. This will leave the top of the cartridge projecting by about ½cm. The Microdrive and cartridge are now ready to be used.

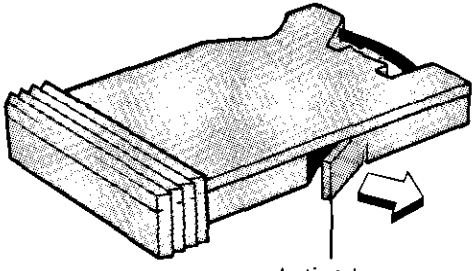
When you have finished using the cartridge, pull it carefully out of the Microdrive, and replace it immediately in its box. But remember . . .

NEVER TAKE THE CARTRIDGE OUT OF THE MICRODRIVE WHILE THE MICRODRIVE LIGHT IS ON.

NEVER SWITCH THE POWER ON OR OFF WHILE A CARTRIDGE IS IN THE MICRODRIVE.

Protecting programs and data stored in cartridges

If you wish to protect the programs and data stored in a cartridge (that is, prevent them from being over-written or erased) then you should snap off the plastic tab from the side of the cartridge, as shown below.



(As with a cassette, you can reverse this effect by sticking some tape over the place where the tab has been removed.)

The cartridge lifespan

Microdrive cartridges will not last forever, and will eventually need to be replaced. The symptom of an ageing cartridge is that the computer will take longer and longer to find a program or file before loading it. So it is a good idea to keep back-up copies of important programs and files either on another cartridge, or on a cassette.

Starting with your Microdrive

Auto-run

Now that you have set up your ZX Interface 1 and your Microdrive, you will want to know what programs are waiting for you on the demonstration cartridge. So, insert the cartridge into the Microdrive (or, if you have several Microdrives, into Microdrive 1), and enter:

NEW

followed by:

RUN (and **ENTER**)

This will automatically load and run the first program on the cartridge. When you have finished watching this program, read on.

The catalogue

To find out what other programs are stored in the demonstration cartridge, enter the CATalogue statement:

CAT 1

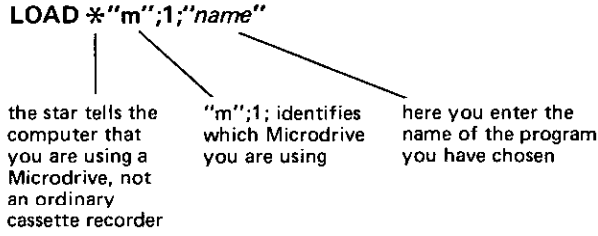
1 identifies the number of the Microdrive you are using

In about seven seconds the television screen will display:

- the name of the cartridge,
- an alphabetical catalogue of the names of all the files stored in the cartridge,
- the amount of memory left in the cartridge (in kilobytes).

Loading programs

The next thing to do is **LOAD** whichever program you would like to see next. So, first choose your program, then enter:



After a short pause, the screen will display the OK message (though not the name of the program). You can now run the program.

Programs and the Microdrive

Saving, verifying, loading and merging programs

In the ZX Spectrum Introduction booklet you found out how to **SAVE** programs on a cassette tape. Saving programs in cartridges is just as easy. Here again is a program called Squares, much like the one in the Introduction booklet:

```
10 REM Squares
20 FOR n=1 TO 10
30 PRINT n,n * n
40 NEXT n
```

To **SAVE** this on a cassette tape you would enter:

```
SAVE "Squares"
```

To **SAVE** it in the cartridge in Microdrive 1, you enter instead:

```
SAVE *"m";1;"Squares"
```

After about six seconds the border will flash, and the program will be saved. Try it and see.

(The names of programs stored in a cartridge, like those of programs stored on a cassette tape, can be up to ten characters long.)

As you would expect, you can now check that the program has been saved by entering:

```
VERIFY *"m";1;"Squares"
```

The screen will then display the OK message.

You can now **LOAD** the program Squares by entering:

```
NEW
```

followed by:

```
LOAD *"m";1;"Squares"
```

then:

```
ENTER
```

Next, to make the program run automatically, try entering:

```
SAVE *"m";1;"Squares 2" LINE 10
```

then:

```
NEW
```

and then:

```
LOAD *"m";1;"Squares 2"
```

Programs can also be merged using a Microdrive. Enter:

```
NEW
```

followed by:

```
100 REM more Squares
110 FOR n=11 TO 20
120 PRINT n,n*n
130 NEXT n
```

And now enter:

```
MERGE *"m";1;"Squares"
```

then:

```
ENTER
```

The program Squares will be added to the listing.

In short, as you have probably already realised, all the syntax used within an ordinary cassette interface (explained in chapter 20 of the BASIC programming manual) applies also to the Microdrive.

Erasing programs

Suppose you have just finished with the program Squares. To erase it, enter:

```
ERASE "m";1;"Squares"
```

(As before "m";1 simply indicates which Microdrive you are using.)

During the **ERASE** statement the border will flash.

If you **BREAK** while saving a program then you will have an unclosed file in the cartridge. You cannot **LOAD** an unclosed file, and an attempt to do so will give the report **file not found.** The **ERASE** Statement can delete an unclosed file, but will take about thirty seconds to do so because the computer checks the cartridge several times to make sure that the file has no end.

Formatting and naming blank cartridges

If you have bought a blank cartridge, before you can use it you must insert it into a Microdrive (say, into Microdrive 1) and enter this:

FORMAT "m";1;"*name*"

"m";1 identifies the Microdrive you are using (in this case, number 1)

enter here whatever name you wish to appear in the cartridge catalogue. It can be up to 10 characters in length

The computer takes about thirty seconds to format a cartridge. During this time the border will first flash, then clear, then flash again, and finally display the OK message. What the computer is doing is identifying any areas that cannot be written to or read from and marking them to be avoided.

Formatting need never be repeated. Note, by the way, that when you format a cartridge, any information it contains will be erased. So, if you have bought several Microdrives, you can **FORMAT** your spare demonstration cartridges and use them as blank cartridges.

Next type:

CAT 1

1 identifies the number of the Microdrive you are using

In a few seconds, the television screen will display the cartridge's name and its capacity. The capacity will vary from cartridge to cartridge, but it should never be less than 85 kilobytes.

Setting up your own auto-run facility

Earlier, you used the auto-run facility on the demonstration cartridge. If you have a program that you often use, you can set up your own auto-run facility, so as to avoid repeatedly having to type the **LOAD** and **RUN** statements. These are the rules to follow:

- the program must be called **run**;
- the cartridge must be used in Microdrive 1;
- the facility must be used either immediately after switching on, or immediately after entering **NEW.**

So, enter the required program, followed by:

1Ø SAVE *"m";1;"run" LINE *number*

the name **run** must be typed out in full. Do not just press the **RUN** key

enter here the appropriate line number

Now enter:

NEW

followed by:

RUN (and **ENTER**)

enter the keyword, not the file name

Note, however, that the **MERGE** statement does not work with any program saved using the **SAVE*... LINE ...** statement. (This is so as to protect such programs.) An attempt to **MERGE** will give the report **'Merge' error.**

Data, channels and streams

As you know, a program is a set of operations which is executed when you type **RUN**. Data, on the other hand, is any collection of letters, numbers or symbols on which a program might operate. An example is the numbers 1 to 10 with their squares.

Data can be sent to, and come from, various parts of a computer system. Such parts are known as *channels*. The channels you can send data to are:

- your screen,
- a ZX Printer,
- a Microdrive file, to store it,
- another ZX Spectrum computer, if both computers are on a network.
- the RS232 Interface and from there, for example, to a modem or printer.

And the channels from which data can come are:

- the keyboard,
- a Microdrive file,
- another ZX Spectrum computer, if both computers are on a network,
- the RS232 Interface, a modem or a terminal.

The routes along which the data flows to and from these channels are known as *streams*. In the Spectrum computer system the number of these streams is fixed at sixteen. They are numbered from 0 to 15, and *stream numbers* are always preceded by the sign #.

Four of these streams come already linked to channels:

stream #0 stream #1	*output* data to the lower part of the television screen and *input* from the keyboard;
stream #2	*outputs* to the upper part of the television screen, but cannot *input*;
stream #3	*outputs* via the ZX Printer; but cannot *input*.

Every statement that produces input or output uses one of these streams automatically. For example, the **PRINT** statement uses stream #2, and the **LPRINT** statement uses stream #3. So, if you enter:

PRINT "This is a Spectrum computer"

it is, in fact, shorthand for:

PRINT #2;"This is a Spectrum computer"

Try entering both, and see.

You can, however, make any statement use a different stream by keying # followed by the new stream number. Try entering:

LPRINT #2;"This is a Spectrum computer"

Instead of being printed by the ZX Printer, this line too appears on the screen.

But as well as using the *established* stream-channels, you can create some of your own. Streams #4 to #15 are free for this purpose; and there are various *channel specifiers* which enable you to indicate which peripheral you require. Some of these are:

"K" for the keyboard

"S" for the screen

"P" for the ZX Printer.

(You will be introduced to the others later.)

Note that **K, S** and **P** are all established channels. They require you to use commas (,) as separators in **OPEN #** statements. But with other channels you can use either commas or semi colons (;).

To create your own stream-channels you use the **OPEN #** statement. For example enter:

10 OPEN #4,"S"

You will have opened stream 4 and linked it to channel **S.** Now enter:

20 PRINT #4; "This is a Spectrum computer"

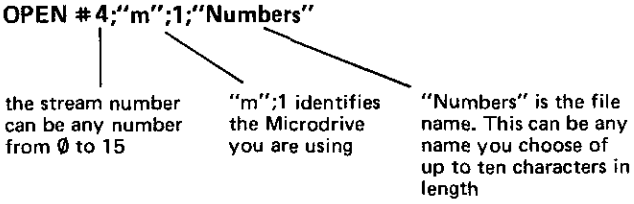
Again, the line will appear on the screen.

(It is not advisable to **OPEN** to streams **0, 1** or **2** as the results are unpredictable.)

Data and the Microdrive

Opening and naming a data file

When you store information in a cartridge you keep it in a file. You also give the file a name so that you can find it again later. The statement that opens and names a data file always takes the same form. For example try entering:



This statement does two distinct jobs:

– it sets up a new channel: "m";1;"Numbers"

– it attaches this new channel to stream #4.

This will have taken about seven seconds, during which time the computer searches the cartridge for a file called "Numbers". Since there is no file "Numbers", it opens a file for writing. (However, if it *had* found a file "Numbers", it would have opened it for reading. And had it found a *program* "Numbers", it would have given the report **Wrong file type.**)

Entering data

Once you have opened a file you can enter data. Suppose that you want to store the numbers 1 to 1Ø with their squares. Enter and run this:

```
1Ø FOR n=1 TO 1Ø
2Ø PRINT #4,n'n*n
3Ø NEXT n
```

You might think that all the numbers have now been stored away in the cartridge. But in fact the computer does not automatically transfer anything to the cartridge until a certain amount has built up, which it transfers all at once. This is called *buffering*. A Microdrive buffer is 512 bytes (or characters) long.

To store in the cartridge the data you have entered you must **CLOSE** the file. Until this is done you will be unable to read back from the file.

Closing a file

Closing a file ensures that the file is safely stored in the cartridge. It also closes the channel (in this case "m";1;"Numbers") and leaves the stream (in this case # 4) with no channels attached. To **CLOSE** a file you need only **CLOSE** the appropriate stream. So enter:

```
CLOSE # 4
```

The border will then flash to show that something is being stored in the cartridge.

(Note that, like the **OPEN** statement, the **CLOSE** statement is followed automatically by # .)

You cannot, by the way, **CLOSE** streams # Ø, #1, # 2 or #3. If you try to do so, streams # Ø and # 1 will default to channel K; stream # 2 will default to channel S; and stream # 3 will default to channel P (see page 22).

Reading back from a file

To read back from the file "Numbers", run this:

```
1Ø OPEN # 4;"m";1;"Numbers"
2Ø FOR b=1 TO 1Ø
3Ø INPUT # 4;m;n
4Ø PRINT "The Square of ";m;" is ";n
5Ø NEXT b
6Ø CLOSE # 4
RUN
```

leave spaces here

(Note at this point that because the file "Numbers" already exists, the channel "m";1;"Numbers" is opened for input, and trying to output it would give an error.)

You can also use **INKEY$** to read back from a file (it always gives the next character in the file). Try this program:

```
1Ø OPEN # 11;"m";1;"listing"
2Ø LIST # 11
3Ø CLOSE # 11
4Ø OPEN # 12;"m";1;"listing"
5Ø PRINT INKEY$ #12;
6Ø GO TO 5Ø
```

This will finish with the **End of file** report.

Notes on PRINT and INPUT

Because the **PRINT** and **INPUT** statements are designed mainly for use with the screen and keyboard, you must take care when using them with files.

separators

The **PRINT** statement uses three forms of separator:

- the ; (semi colon) prints nothing,
- the , (comma) takes you to the start of the next half line,
- the ' (apostrophe) gives a new line (the **ENTER** code).

The **INPUT** statement always expects you to type **ENTER** after a number or a string. So, when you are printing to any file from which you expect to **INPUT**, you must either:

- print the items singly, e.g.

```
10 PRINT #4;2
20 PRINT #4;3
```

or

- separate them with an apostrophe, e.g.

```
10 PRINT #4;2'3
```

You must also take care when using separators in an **INPUT** statement. As you know, **INPUT** can print to the bottom half of the screen anything that you can put in a **PRINT** statement. But when you **INPUT** from a file, the file is only open for reading. So, if you include anything that would be printed when using the screen, you will get the error report **Writing to a 'read' file.** This means that items in the **INPUT** statement should be separated with a semi colon, e.g.

```
10 INPUT #4;a;b
```

Be careful also when you **INPUT** a string containing " (quotes), because the **INPUT** will think that the " is the end of the string. The way round this is to replace, for example:

```
10 INPUT #4;a$
```

with

```
10 INPUT #4; LINE a$
```

changing streams

PRINT statements may also contain information for several streams at a time. The following program will print "one" on the screen; "two" to a Microdrive file called "digits" in Microdrive 1; "three" to station 1 on a network (see chapter 7); and "four" to the next line on the screen.

```
10 OPEN #4;"m";1;"digits"
20 OPEN #5;"n";1
30 PRINT "one"; #4;"two";#5;"three" '#2"four"
40 CLOSE #4
50 CLOSE #5
```

setting colours

After using a channel other than the screen, you may find that the **PAPER** and **INK** statements have no effect. To get round this, enter:

```
PRINT;
```

before setting **PAPER** and **INK**.

For more about **PRINT** and **INPUT** see chapter 15 of the Basic Programming manual.

Reading the file catalogue

As you establish files in a cartridge they are automatically catalogued. So, when you want to find out what files a cartridge contains, you need only insert the cartridge into a Microdrive and enter the CATalogue statement. For example, enter:

CAT 1

The television screen will display:

- the cartridge name,
- the file names,
- the amount of memory left in the cartridge, (in kilobytes).

You can also send the output of **CAT** to a stream by entering:

CAT # *number;number*

enter the appropriate stream number here

enter the Microdrive number here

This enables you to send the catalogue to a printer, or to a file, so that a