ALLAN SCOTT

THE SPECTRUM ADD-ON GUIDE

The Spectrum Add-on Guide

Allan Scott

GRANADA
London Toronto Sydney New York

Granada Technical Books
Granada Publishing Ltd
8 Grafton Street, London W1X 3LA

First published in Great Britain by
Granada Publishing 1984

British Library Cataloguing in Publication Data
Scott, Allan
The Spectrum add-on guide.
1. Sinclair ZX Spectrum (Computer)
2. Computer interfaces
I. Title
001.64'4 QA76.8.S625

ISBN 0 246 12563 2

Typeset by V & M Graphics Ltd, Aylesbury, Bucks
Printed and bound in Great Britain by
Mackays of Chatham, Kent

Contents

Acknowledgements

My thanks first to the many companies (including, of course, Sinclair Research) who provided samples and information about their products so promptly and efficiently – and a special word for Steve Money, who stepped into the breach at a moment of crisis to lend me a second 48K Spectrum. The massed arcade players of Watford gave me an unforgettable evening and the valuable rating table in Chapter 2; I hope they enjoyed it as much as I did. My thanks also to Richard Miles for suggesting the project, to Ann Berne for assistance in research far above the call of duty, to Sue Moore for her usual speedy and efficient processing of the manuscript, to Mike Scott Rohan for most of Chapter 10 and Appendix 2, plus reams of advice and information, and to Phil Gardner who wrote Chapter 8 and much of the programming for the main text routines.

Important Note

There are a few simple rules for using add-ons with your Spectrum. Please read this section carefully: it could save you a lot of money, a lot of time, and a great deal of fruitless argument with manufacturers!

- *Always* make quite certain that power to your Spectrum has been switched off at the mains before connecting or disconnecting any add-on unit. Failure to do so could damage both the computer and the add-on.

- Before plugging in an add-on unit, check that the *key* (usually a small piece of metal) is in place on the edge connector (see Fig. 1.1). If the key has fallen out and a connection is made, damage could result.

- When connecting equipment of any kind to the user port at the back of the Spectrum, make sure that the key lines up with the slot visible on the printed circuit board inside the user port (see Fig. 1.1).

- Some of the equipment mentioned in this book was originally designed for use with the ZX81. Edge connectors on equipment of this type are noticeably shorter than the matching circuit board in the user port, and you should take particular care when lining them up for connection. Check when first setting up equipment of this type that no converter circuitry is needed. Some units (e.g. the ZonX sound unit reviewed in Chapter 6) are supplied with extra connecting boards so they can be fitted to the Spectrum.

- Many add-on units have no rear connector for the addition of further units – they have to be 'the last in the chain'. You can beat this problem (at extra cost) by using cables and connectors available from good computer shops. This also solves the problem of linking up units with ZX81 and Spectrum connections, but it's expensive, untidy, and inconvenient, and units that cause these difficulties do correspondingly less well in the rating tables later in this book.

- Don't overload the back of your Spectrum with add-on units. It is quite possible that two or more of them could be competing for the same area of memory, and if this happens the computer may well 'lock up': the screen will go blank, or a fixed pattern will appear, and no key will operate. If this happens to you, turn off the power *at the mains* (not at the back of the computer), disconnect one add-on unit at a time, and try to find which units are in conflict. Incidentally, a lock-up only *looks* serious – all the same, it does mean you will lose any data that was in the computer at the time.

- Prices and technical details were correct at the time of going to press, but obviously these are subject to change in this highly competitive market. Treat them as a guide rather than gospel!

Introduction

This is not the only book about Spectrum add-ons on the market. Nor will it be the last. Unlike some books, however, you won't need a degree in computing science to understand it. It's written in English for ordinary Spectrum users who want to improve their system – and if you just want to know which joystick is the best value for money, you'll find the answer here.

I won't pretend that this is a complete guide, because no book can be. New equipment is always appearing, and firms do sometimes go out of business. However, when this book went to press every manufacturer I had been able to trace had been contacted, and all had been invited to submit samples of their product.

In every chapter I have tried to answer four basic questions about the product under review: what it is, how it works, what it does, and what *you* can do with it. Wherever possible, each product has been tested by people who, like you, have to rely on the instructions and programs the manufacturers give them. Sometimes those instructions are extremely poor: if so, we have said so, and in some cases tried to fill in the gaps for you if the product itself is worth the effort. We have also put together some programs for the hardware of our choice, to demonstrate a few of the things it can do and give you some ideas for your own routines. After all, you can't be expected to know what a piece of equipment you have never seen is and does.

To make life more interesting for you, many of these add-ons can also be used to play the Spectrum adventure game in Appendix I – you don't actually need the add-ons if you don't have them, but you'll enjoy the game that much more if you do. We hope this will give you something to do when you get tired of reading about hardware!

First-time users start here ...

If you have just bought, or are just about to buy, a ZX Spectrum microcomputer, then you may find the idea of add-ons rather puzzling. Why, after all, should you want to add anything on to a computer that already has such an enviable reputation for features and memory size?

The ZX Spectrum *is* a remarkable computer. It is based around an excellent processor chip, the Z80A, which makes it fast and versatile. It has more memory available to the user than several machines at twice the price. It has high-resolution graphics, and an ingenious colour system that allows elaborate screen displays without eating up vast areas of memory. The Spectrum was the first real computer to be sold at a price that most people could afford – and I stress that word 'real'. And as if that weren't enough, the software manufacturers have been falling over each other to bring out programs for it: programs that can help you with anything from drawing up your accounts to drawing pictures of hobbits at bathtime.

So, who needs add-ons? You do – in fact you are using at least two of them already. Your TV set and your cassette recorder are both add-ons (the proper term, incidentally, is *peripherals*) and without these peripherals you'd be pretty helpless. The computer doesn't need the TV set – it will function quite happily without it – but *you* won't, and that's the main reason for buying any peripheral. A useful peripheral is one that helps you do a particular task more easily. If it doesn't do that then you're wasting your money. And as to the sort of tasks that peripherals can be used for – well, you're looking at one.

Ladies and gentlemen, a demonstration ...

This book was typed using a ZX Spectrum fitted with a Fuller FDS keyboard and a Cub monitor and running the excellent Tasword 2 word processing program. The completed files were stored on Microdrive cartridges – when I'd finished with them I used a Prism VTX 5000 modem and OEL's new user-to-user software to send their contents down the phone line to Phil Gardner for copy-editing. (He lives in Leeds, and I'm 300 miles nearer London, in Watford.) At the other end, Phil was able to copy-edit my work on *his* Spectrum and then print out the results on his Lucas printer using a Hilderbay interface. Meanwhile, much of the artwork for the book was being prepared on my Spectrum using the RD Digital Tracer, and I was printing out the results, and a copy of the draft manuscript, using the Morex interface and a Kaga Taxan printer. Effectively, this whole book is a product of Spectrum add-ons. If you don't feel ready to write your first book at the moment, then don't worry. That certainly isn't all the Spectrum can do.

If you have owned a Spectrum for some time then you probably know quite a lot about its capabilities, and you may already know how much more is possible. In the chapters that follow you'll find a great deal of information about all those tempting peripherals that appear so regularly in the computer press. If you're a newcomer to computing then this book will introduce you to a Spectrum you may never have known existed. It can become a high-speed games machine. It can produce high-quality printing on anything from a personal letter to a full-length novel, using a comfortable custom-designed keyboard. It can access hundreds of kilobytes of memory in seconds (if you're not sure what kilobytes are, take a quick look at Chapter 1). It can talk to you, sing to you, play three-part harmonies for you, and help you to compose the next number one hit single. It can be used to copy photographs, drawings, original art, and your own screen designs. It can access huge banks of data on mainframe computers for little more than the cost of a phone call, and let you 'talk' to almost any other computer. It can even control household equipment – anything from calling you up at the office to tell you the house is on fire to operating a domestic robot.

Like any good computer, the Spectrum is a sort of ideal workman. If you give it the right tools and the right instructions, it will do almost anything. Why? That's a question I'll try to answer in the next chapter – and then we'll take a detailed look at the 'tools' that are available, and the jobs they can do.

Chapter One

What does it do? Well, what do you want it to do?

What exactly is a computer?

No, it isn't a trick question. The point is that you won't find it very easy to understand what the hardware in the next few chapters is and does unless you have a clear idea of what you're plugging it into. So even if you're sure you know exactly what a computer is, bear with the rest of us for a paragraph or two.

First of all, let's consider what a computer isn't. To start with, it isn't a thinking machine, as you will already have found if you've been trying to program it! On the other hand it isn't just a device for rearranging numbers and data, either. Getting right back to fundamentals, a computer is an organised network of electronic switches. Each switch can be set either 'on' or 'off'; or, if you like, at '1' or at '0'. The Spectrum (like every other home computer) organises these switches, known as *bits*, in groups of eight, known as *bytes*. Modern computers have so many bytes available that it's convenient to count them in *kilobytes* (K for short). A kilobyte is 1024 bytes. This may seem a peculiar number, but the computer finds it easier than 1000 because it counts in twos and powers of two, and 1024 happens to be 2 to the power of 10. Each byte has its own 'telephone number' (the correct term is *address*), and the 48K Spectrum has a total of 65536 of these addresses available to it. You may already have noticed that 65536 is equal to 64 × 1024, which is 64K! So why is it called the 48K Spectrum?

The short answer is that every one of those extra 16K addresses is used, but none of them is available to you. That's because the computer needs them for its BASIC, its normal housekeeping operations, and everything else that makes it an active machine rather than a passive box of switches. In other words, the only part of the memory you can actually use for storing data or programs is the part that's left over when the computer has taken what it needs.

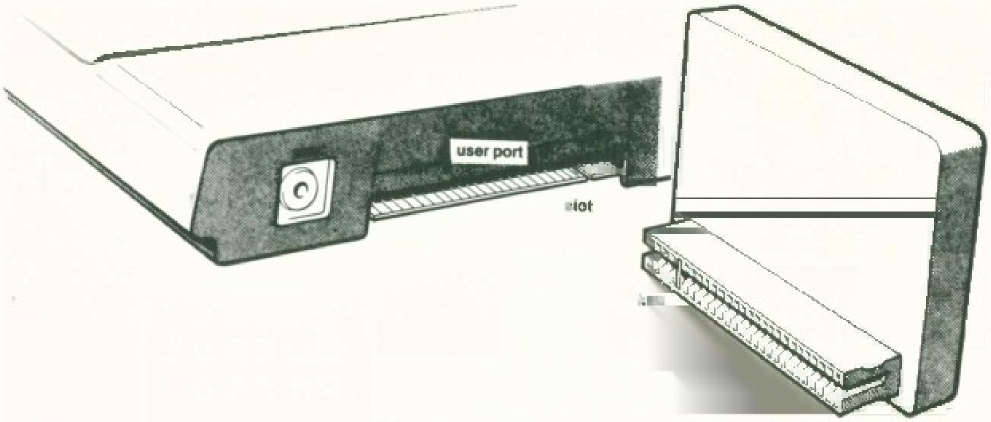

Fig. 1.1. Spectrum user port, and a typical add-on showing edge connector.

Naming of parts . . .

The section of memory used by the computer is called ROM, which stands for Read Only Memory – in other words you can't 'write' anything into it. The section of memory you *can* 'write' into is called RAM, which stands for Random Access Memory, or, if you like, 'Read And write Memory' (not an official description!). The part of the computer that does all the work actually lies outside all the memory circuits, as you can see in Fig. 1.2. It's called the CPU, or Central Processor Unit, and it's based around the Z80A microchip. You can think of the CPU as the Spectrum's telephone exchange. It can receive information from one 'telephone number' in memory and then send that information to another 'telephone number' somewhere else in memory. If something more complicated is required – compound arithmetic, for instance – then it goes to ROM and gets the necessary instructions. Besides being a message-carrier, the CPU can also perform simple arithmetic, and compare simple pieces of information. That's about all. It may not sound very impressive, and indeed it isn't; what makes the chip so useful is that it can perform these simple operations up to 3500000 times a second.

So, to put it at its simplest, the CPU takes information that you, the user, have put into RAM, sorts it out, and sends the results wherever it has been told to send them. This brings me to the point of all this explanation. The CPU isn't bothered where its information comes from, or what happens to that information once it has been sent out. It can even come from outside the computer, and go somewhere else outside the computer when the CPU has finished

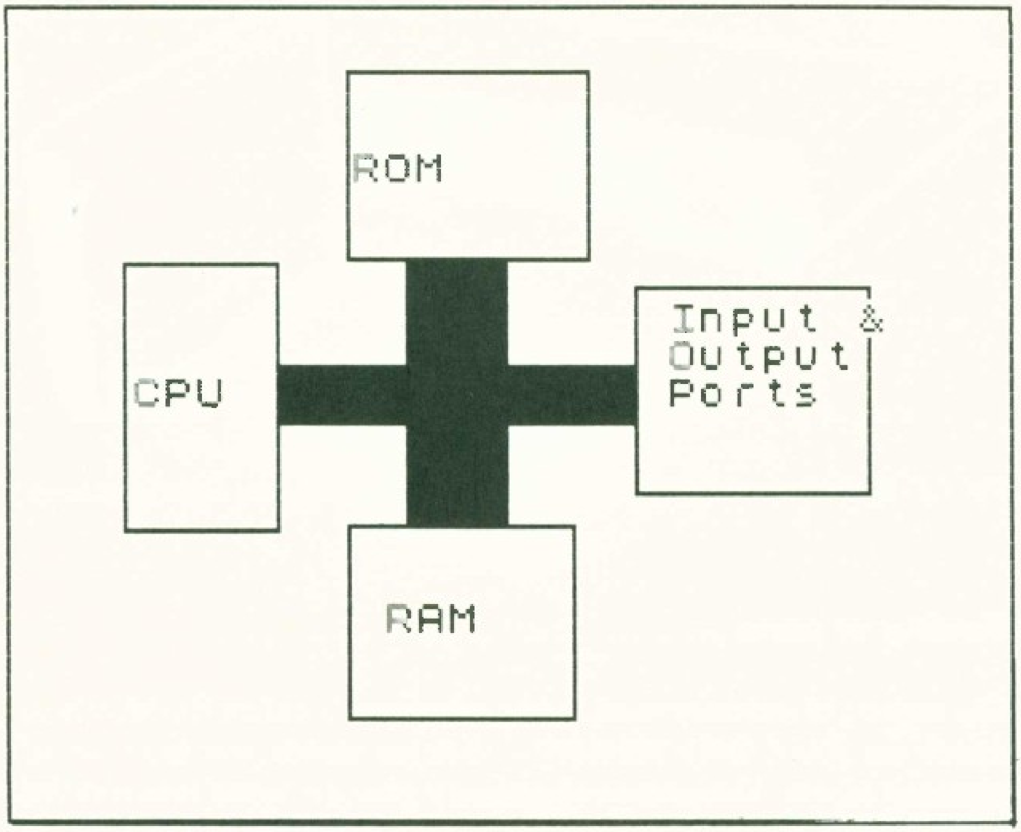

Fig. 1.2. Block diagram of the Spectrum (created with the RD Digital Tracer).

with it. You can see this for yourself every time you LOAD or SAVE a program on cassette! Instead of sending bytes to a different address in memory, your Spectrum simply sends them to *output ports*. The difference has no significance for the CPU, but quite a lot for you! It means that your cassette recorder can now be fed with the individual codes that make up a program, byte by byte, and record them as audible signals on tape. There's more to it than that, but the basic idea is important. Ports are the CPU's link to the outside world. The keyboard is connected to it by ports, and so is the TV. But there are plenty of spare port addresses – and, as you'll see, plenty of equipment you can connect through those addresses!

Input and output

Any device that uses the input and output facilities of your Spectrum, from a cassette deck to a household robot, operates on the same simple principles. As far as the computer is concerned, the information it is receiving from these devices and sending out to them is just a series of switch positions coming from its normal addresses.

If this puzzles you, then try to imagine what actually happens when you 'program' your Spectrum. Remember that the CPU can only work with eight-bit codes, so it certainly doesn't 'understand' the BASIC keywords you are using to program it. In fact, each of those keywords is stored in RAM as a number code, or *token*. If, for instance, you key in the command PRINT "a" (see Fig. 1.3) the CPU

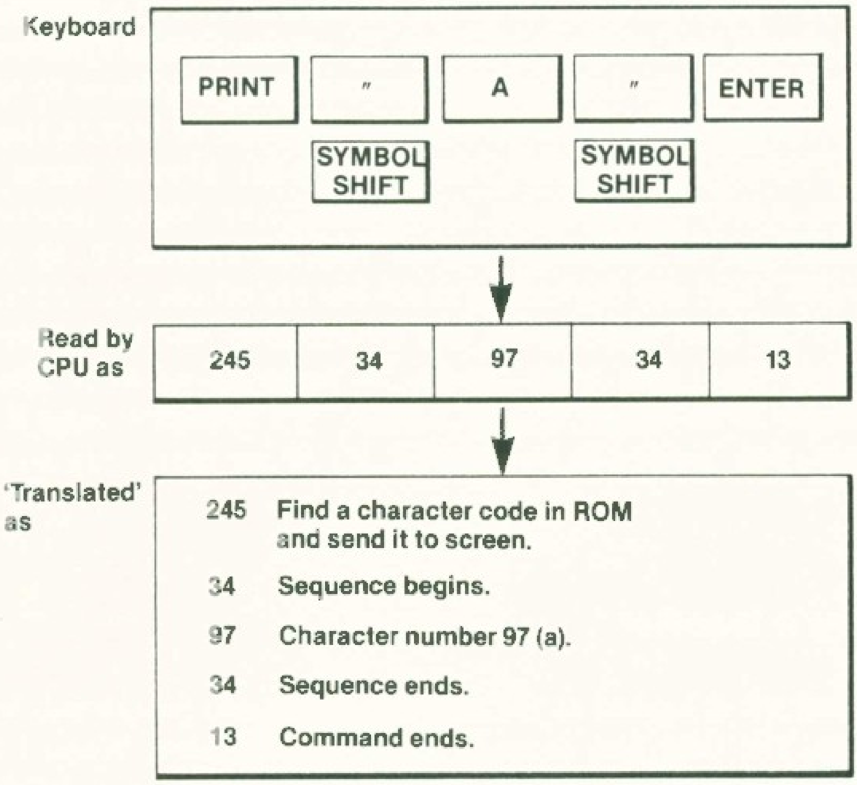

Fig. 1.3. How the computer sees a direct command to PRINT "a".

picks up your command as a series of signals from the input ports connected to the keyboard: signals which come through as a string of numbers. These numbers call up a set of pre-programmed instructions, and also supply the data the computer needs to carry them out. In this case the CPU finds the information about the letter 'a' in ROM and sends it to the output ports that go to your TV. The result is that a pattern of dots and spaces appears on the screen corresponding to the pattern of 'on' and 'off' switches that was originally stored in ROM. There's a bit more to it than that, but as far as the computer is concerned, all it has done is to transfer information from one part of memory to another.

Using input and output

At the moment you only have a limited number of things to attach to the output and input ports: your cassette deck, your TV set, and your printer (if you have one). But as you look through this book you'll see many other devices, including those mentioned in the introduction. Now you know something about the basic principles they should be looking a little less mysterious! Digital tracers and lightpens simply convert drawings on paper or movements across a TV screen into numbers – numbers that can be used by a program to create shapes and outlines on your TV screen. Sound synthesisers take number codes from your Spectrum and use them to make sounds: noise, music, or a close approximation to human speech! Robots simply convert code numbers into actual movements. But even this is only the beginning. Any device controlled by switches or by variations in electrical current can be controlled by your computer – and any device that sends out on/off codes or electrical currents can pass instructions and information into it. In fact, the computer that runs your house in response to your spoken commands is no longer science fiction – it's available. And if, for instance, you happen to be disabled, it's rather more than a clever toy.

Getting it together

Now that we've looked at the way computers and peripherals work together, the time has come to speak of interfaces. When I first became interested in computers I was convinced that the word 'interface' was a sales gimmick: an attempt to make me pay an outrageous price for a common or garden plug. But usually an interface is a lot more than 'just a plug', and now perhaps you can see why it has to be. An interface has two very important jobs to do:

- To translate information sent by a peripheral into codes the computer can understand.
- To translate codes from the computer into information the peripheral can understand.

In other words, if the CPU is the telephone exchange then the interface is the translator which makes sure that the computer and the peripheral understand each other.

In practice this means different things for different types of equipment. The 'interface' for your TV set looks, at first sight, very simple. All you can see is the socket for your aerial lead. What you can't see is the whole system of memory layouts and the special video chip that translate an ordinary set of computer codes into something that can be shown on a TV screen. All that is 'built into' your Spectrum, but it's as much a part of the TV interface as that socket. An interface is often more than a piece of hardware. It may include *software* – programmed instructions – too. This software can be part of the interface itself (rather like an extension to the Spectrum ROM, in which case it's known as *firmware*) or it may be a program you have to load into the computer before you can use a particular device. Either way, there's usually a lot more to using a peripheral than just slotting it in at the back! Some peripherals are almost like small self-contained computers. Speech synthesisers, for instance, usually have built-in circuits to translate codes from the computer into electrical impulses that can drive a speaker.

Getting down to brass contacts

That's enough theory for the moment! It's time to take a proper look at the different types of peripheral available for the Spectrum, and see just what they can do to help you. Each of the chapters that follows is pretty much self-contained: so if you're mainly interested in playing games, Chapter 2 should be right up your alley, and if you're a budding author who thinks computer games are for kids you can go straight to Chapter 8 (though I'd suggest you try a couple of games before jumping to conclusions!). Even if you think a particular chapter will not help you much, take the time to skim through it, and do read the sections dealing with applications. A peripheral intended for games players may well have a serious business application, and another intended mainly for business users may help you create some superb arcade games. But for the young among you (and the young at heart) let's start with the games.

Chapter Two
Joysticks and Joystick Interfaces

It's perfectly possible to play arcade-type games on the Spectrum using the keyboard alone – but it isn't particularly easy. You may have bought Spectrum as a 'serious' computer, but if certain users of any age from six to sixty get their hands on it, it will soon become a private arcade machine! I see nothing wrong with this at all. If I'd been able to develop eye and hand coordination in such an entertaining and painless way I might have made more of a showing on the cricket field (but probably not!). Anyway, before long even the arcade games are going to be educational. ('You are Napoleon at the battle of Waterloo. Use the joystick to rearrange your forces and defeat Wellington ...')

If you feel that joysticks are 'just for games', it's worth remembering that joysticks are a simple and convenient way of manoeuvring *anything* around the screen, including cursors in word processing programs, graphics programs and large menus. You can even use them to help edit your own BASIC programs, and I use a joystick and programmable interface to handle some of the harder-to-reach Tasword commands.

The basic Spectrum is not, of course, capable of handling joysticks directly: there is nowhere to plug them in and no interface to decode their signals. As a result, there's a bewildering number of different interfaces and joysticks available from a wide range of manufacturers. To spare you some of the bewilderment, I can tell you that any joystick with a D-type (Atari) plug, as shown in Fig. 2.1, will normally work with any Spectrum-compatible joystick interface that has the matching socket. I haven't even tried to list the full range here, but this is a representative sample of what's on offer.

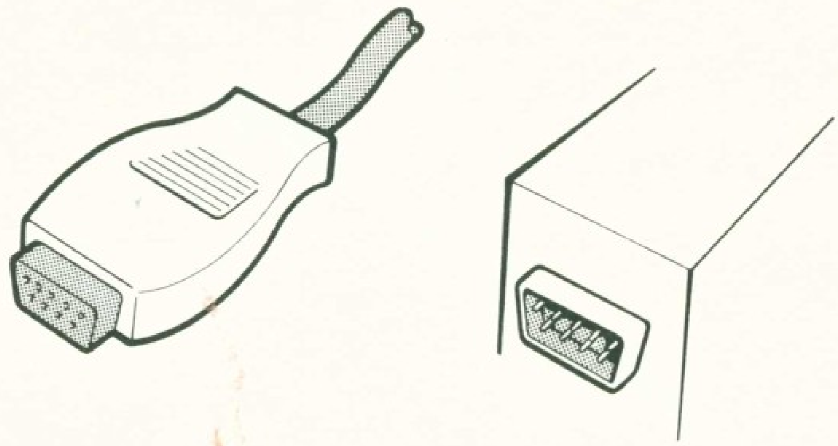

Fig. 2.1. Atari D-type plug and socket.

Joysticks – what they are and how they work

Most joysticks operate on one of two basic principles. The *switch* type just 'replaces' keyboard keys that would normally be used for controlling direction. You move your stick in any of the eight directions shown in Fig. 2.2 and it makes a connection inside the joystick. The interface translates that connection into the code for the equivalent key, or combination of keys, on the keyboard. If you like, it fools the CPU into believing that the key has been pressed.

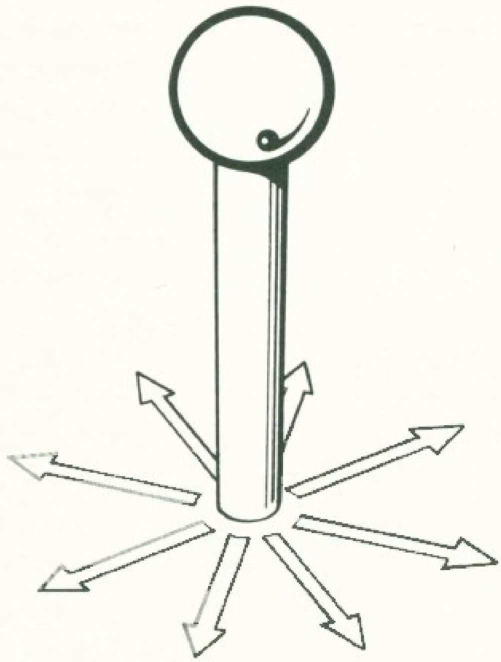

Fig. 2.2. A switch joystick can normally control movement in eight directions.

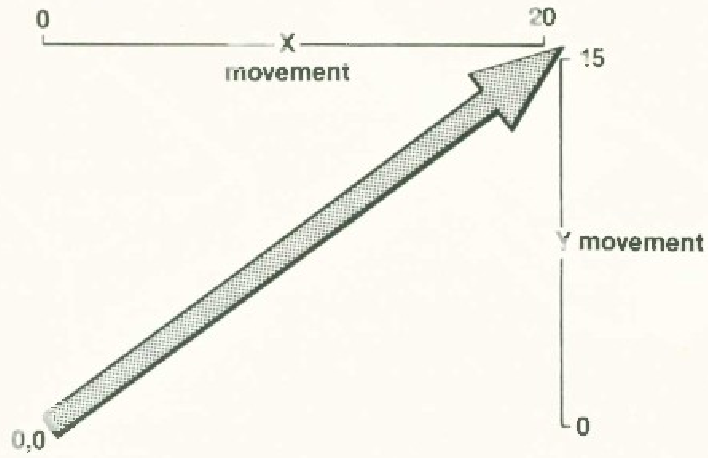

Fig. 2.3. An analogue joystick 'translates' movements into variations in horizontal (x) and vertical (y) positions.

The *analogue* joystick, on the other hand, uses two *potentiometers* (voltage controls – very similar to the one that controls the volume on your radio). Up-down movement alters one potentiometer, and the other is altered by sideways movement. But that doesn't mean you can only move in four directions! Movement in any direction can be broken down into vertical and horizontal parts, and Fig. 2.3 should show you what I mean. Analogue joysticks are perfect for more serious applications, involving high-resolution graphics, or games you want to program yourself, but very few commercially available games are designed for them. However, control is far subtler than with the switch joystick, and certainly far more accurate.

There are a few joysticks that fall outside these categories; broadly speaking these are the gloriously simple and surprisingly effective mechanical joysticks from Grant Design and EEC; the revolutionary Trickstick from East London Robotics; Le Stick, which has no base(!); and the Stack Light Rifle, which could turn your living room into a shooting gallery.

Choosing your weapon

What are you looking for in a joystick? It isn't a silly question – different sticks vary enormously, and a lot depends on what you want to do with them. Check *your* choice for:

- Comfort
- Responsiveness

- Accuracy
- Ruggedness

Comfort is important: any stick that gives you wrist-ache and calluses after half an hour is a bad buy. But comfort's an individual thing – how strong are *your* hands? *Responsiveness* is vital in any application – and some joysticks are decidedly sluggish. Surprisingly enough, a joystick that feels a bit stiff is often the best bet, but it shouldn't be too stiff for you to use over an extended period. It often helps if the stick is *self-centring*, which means that it returns to the centre position after you've moved it in any direction. *Accuracy* has a lot to do with responsiveness – if it's easy to feel when the switching has made a positive connection, it's easier to 'nudge' the joystick into the very small movements required by games like *Splat* (to name but one). Again, self-centring sticks make 'nudging' pretty easy. An inaccurate joystick, however comfortably styled, is not doing its job. *Ruggedness* is essential. Most joysticks will receive a great deal of rough handling, and if they fall to pieces after a few hard wrenches then you have good reason to be annoyed. Take a hard look at any joystick before you buy it, and make sure it will stand up to *your* use of it.

Table 2.1 shows how a sample panel of users rated my selection of joysticks. It is included to give you an idea of how a joystick is *likely* to feel when you first start to use it. Again, I would strongly advise you to try a few out at your local computer shop before making a decision. We all have differently-shaped hands, after all, and one user's ideal joystick could be another's nightmare. And another word of warning – prices on joysticks vary wildly depending on where and how you buy them. Sometimes you do better to buy one with an interface, and it can be worth your while to shop around. Prices given here should be taken as a rough guide. Incidentally, you will find that a number of different suppliers are releasing the same basic joystick – and some don't tell you that theirs is, for instance, the Atari or the Spectravideo. If in doubt, ask!

What the users thought

First choice from the user panel was indisputably the **Spectravideo Quick Shot** joystick. Its large pistol-grip lever is just stiff enough to be positive, but the grip design ensures that even small movements can be made accurately and quickly. There are two fire buttons, one

Table 2.1. Joysticks – what the users thought.

Type	Comfort	Accuracy	Response	Strength	Self-centring	Price
Atari	3	6	6	9	Yes	£7.95
Boss	9	9	9	9	Yes	£14.95
Cambridge Computing	9	9	9	7	Yes	N/A
EEC	8	7	7	8	Yes	£9.95
Grant	8	7	7	8	Yes	£9.95
Kempston Competition Pro	7	7	7	9	Yes	£14.95
Quick Shot	9	9	9	9	Yes	£9.95
Quick Shot II	8	8	9	9	Yes	£16.95
Suncom JoySensor	8	6	5	9	—	£29.95
Superjoy	9	8	8	9	Yes	£7.95
Sure Shot	9	8	9	10	Yes	£13.95
Trickstick	7	4	4	9	—	£34.50
Triga Command	8	8	8	8	Yes	£19.99
Voltmace Delta S3	7	8	8	8	Yes	£10.00

on the top of the stick and one on the base at the left-hand side. You can hold this stick in your hand or use the suckers on the base to attach it firmly to some convenient table top, leaving your other hand free to operate the keyboard commands used in games like *The Alchemist*. It's a good all-round joystick that should stand up to heavy use, and a bargain at £9.95. The **Quick Shot II**, at £16.95, also features a trigger-style fire button but, curiously, the users liked it less.

The **Atari** joystick is almost an industry standard, but it's very stiff and rather uncomfortable to use. It is tough, serviceable, and cheap (£7.95), but certainly not a favourite.

The new **Boss** joystick, just in from the States, is a strong contender for the Best Joystick award, but I miss the Spectravideo's suckers. The trigger grip on this one rotates around the lever – a nice touch. The fire button is on the top of the stick.

Cambridge Computing supply an excellent switch-control joystick with their Intelligent Interface (see below). It has a very positive switch action with audible clicks from the microswitches and a neat self-centring lever. There are two fire buttons. It is only sold with the Intelligent Interface.

The **Kempston** joystick range is aimed largely at the games market, but the panel found these large, chunky lever-type sticks with their big red fire-buttons fairly responsive and hard-wearing. The Kempston sticks are a little stiff, but these are good, solid, basic joysticks at sensible prices, and most of them feature a duplicate fire button for southpaws – or right-handers with tired trigger fingers. The cheapest is the **Competition Pro** at £14.95.

The **Suncom** joystick range includes the first touch joystick, the JoySensor, attractively styled in light grey with three fire buttons (two ordinary ones to right and left and a 'rapid fire' button at the centre). As the stick requires virtually no strength at all, its uncomfortable-looking squared-off shape is perfectly acceptable, but it takes a deal of getting used to and the user panel weren't enthusiastic. Make sure you like this one before you buy, especially at the asking price of £29.95. More conventional is their curiously-shaped but pleasant-to-handle Kraft joystick at £18.95 and the stubby but equally pleasant Starfighter at £13.95.

The **Superjoy** is a cut-down Quick Shot, complete with suckers. The price is cut-down, too!

The **Sure Shot** is a stubby, ex-arcade machine stick with two nice, positive fire buttons and a pleasant click-switch action. I liked it.

The **Triga Command** from the USA is a neat idea – the big pistol-grip lever incorporates a trigger-style fire button – but takes a little getting used to. It features suction cup feet similar to the Spectravideo, but there's no duplicate fire button.

The new **Delta S3** from **Voltmace** is worth a look. It features three fire buttons – the ones on either side are the normal type and the middle one's for 'rapid fire'. It has a pleasant self-centring action, a neat but solid switch-type control, and an attractive and reasonably comfortable styling. A good buy, and good value.

From **EEC** and **Grant Design** come two similar and remarkably effective joysticks that have no electronic parts at all. They simply clamp over the keyboard and use a standard joystick stalk to operate the cursor keys. Although they can only be used on the basic Spectrum keyboard, and with programs that use the cursor keys, they're a bargain at £9.95 if you don't feel that you're going to need a joystick all that often. The design is good, and they're surprisingly robust.

The **Trickstick** from East London Robotics is a stubby cylinder fitted with six red knobs (see Fig. 2.4). The first surprise is that it has no moving parts. The second is what it can do without them, as shown in the demonstration program supplied. With the Trickstick you can quite literally run rings around the opposition on screen.

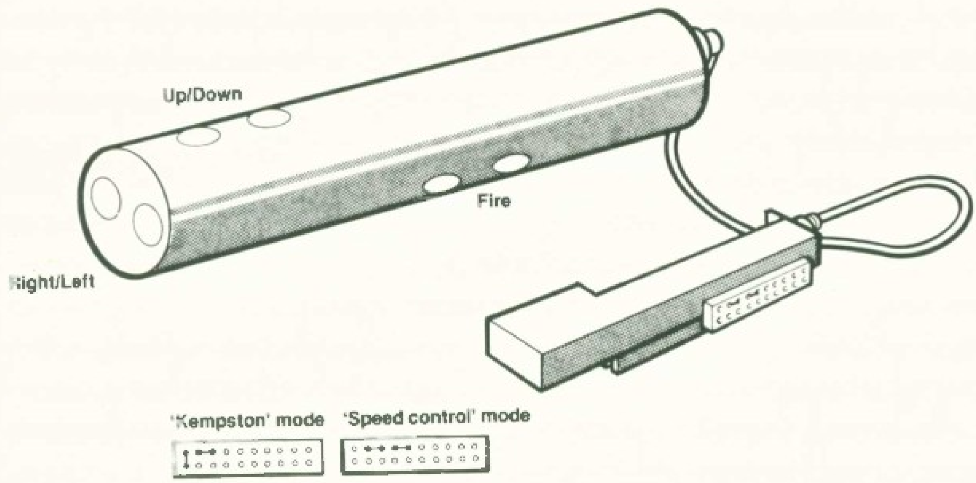

Fig. 2.4. The Trickstick and interface, with pin settings for Kempston mode and for its own software.

This ability to move graphics in smooth circles and curves under full control is a real innovation, and one I hope to see more of. It's also possible to plug up to eight of them into one Spectrum (sounds a bit crowded to me). The stick, so I am told, is powered by a capacitative effect of the human body – apparently all God's children generate a 50 Hz hum.

The Trickstick has two major disadvantages. One is that the games to make full use of it have not yet been written. Second is its sensitivity, which is pretty difficult to adjust. This is important, because the Trickstick works by sensing how hard you are pressing its buttons, and if you get too excited you'll go further and faster than you ever intended to. A pair of curiously tiny and fragile-looking miniature plugs on the interface allow the Trickstick to be 'converted' into something like a standard joystick compatible with all games that are suitable for the Kempston interface (of which more in a moment). The same pins can be used to identify which player is which in a multi-user game.

The software provided by East London Robotics with my Trickstick was slightly odd. There were two tapes, a 'Trickstick trainer' and a game called 'Attactics'. Neither is particularly user-friendly, and despite death-grip finger pressure the stick didn't seem to send 'my' aircraft into the same amazingly tight turns as my opponent's'. In one of the main game options I found that it didn't make much difference to the score whether or not I was holding the stick! I can only hope that the software East London Robotics is promising for the Trickstick is an improvement on this performance. The user panel, alas, was not enthusiastic, mainly because the Trickstick made some conventional games almost unplayable. With software support, and some more development work, this could be a remarkable and radical new stick; without them, it really isn't worth its current price of £34.50.

Le **Stick** is an equally curious-looking design, mainly because it manages without a base. To use it, you must tilt it in the direction you want to move! It works, so I'm told, by detecting the movement of mercury inside the stick itself – the mercury is heavy enough to make the necessary contact. I didn't actually look. My brief session with it convinced me that this was another stick that would take a little getting used to.

The **Light Rifle** is not really a joystick, but a reasonably accurate variant on the lightpen theme (see Chapter 7). It arrives looking like something Robert Mitchum used in World War II – could come in useful for frightening burglars! The three programs supplied are fun, but I didn't get one in time to do an in-depth analysis. If you like the idea, buy one!

Introducing the interface

As mentioned earlier, the job of the joystick interface is to turn switch connections from the joystick into numbers – numbers that duplicate exactly the codes that would be sent by the corresponding keys. Joystick interfaces fall into three main groups according to the way they handle the problem:

- 'Fixed' interfaces. These allow the joystick to duplicate one (fixed) set of keys or (like the Kempston) simply send a set of codes that can be used in a program to a selected input port.
- 'Plug' interfaces. These use plugs or clips so that the user can choose which keys he wishes to duplicate.
- Programmable interfaces. These use on-board firmware or programmable chips called PROMs, and allow any joystick movement to duplicate almost any key entry in any mode.

Making the choice

Your choice, again, will depend on what you want to do with the joystick. 'Fixed' interface manufacturers like Kempston try to persuade the software houses to include the necessary code for their interface as an option in popular programs, solving the compatibility problem by diplomacy rather than flexibility. With the interface fitted you can also use joysticks in your own programs (see Fig. 2.5). If a commercial program doesn't include an option for your interface it may often include another where you pick the keys you want to use – so just pick the keys the interface duplicates.

```
  10 LET d=0: LET x=127: LET y=8
8
  20 INPUT "Paper colour? ";p: I
F p<0 OR p>7 THEN  GO TO 20
  30 INPUT "Ink colour? ";i: IF
i<0 OR i>7 THEN  GO TO 30
  40 INPUT "Border colour? ";b:
IF b<0 OR b>7 THEN  GO TO 40
  50 PAPER p: INK i: BORDER b: C
LS
  60 IF IN 31=2  AND x>0 THEN  L
ET x=x-1
  70 IF IN 31=1  AND x<255 THEN
 LET x=x+1
  80 IF IN 31=4  AND y>0 THEN  L
ET y=y-1
  90 IF IN 31=8  AND y<175 THEN
 LET y=y+1
 100 IF IN 31=10  AND y<175 AND
x>0 THEN  LET y=y+1: LET x=x-1
 110 IF IN 31=9  AND y<175 AND x
<255 THEN  LET y=y+1: LET x=x+1
 120 IF IN 31=6  AND y>0 AND x>0
 THEN  LET y=y-1: LET x=x-1
 130 IF IN 31=5  AND y>0 AND x<2
55 THEN  LET y=y-1: LET x=x+1
 140 IF IN 31=16 THEN  LET d=d+1
 200 IF d=1 THEN  PLOT x,y: GO T
O 60
 210 IF d=2 THEN  LET d=0: GO TO
 60
 220 GO TO 60
```

Fig. 2.5. Simple sketching program for Kempston.

Normally you can do this just by moving the joystick up, down, left and right (or whatever) in response to the screen prompts from the program. Some interface manufacturers even supply 'conversion tapes' that will set up the computer to play a game that normally uses a different set of codes. However, this is not going to help you with software that was not originally designed for use with joysticks; for that, you'll need something a little more sophisticated.

'Plug' interfaces are versatile and practical. They're also simplicity itself to program. Just take a colour-coded or labelled set of leads (usually UP, DOWN, LEFT, RIGHT, and FIRE), and fit them to sockets or leads corresponding to the keyboard keys you want to duplicate. However, these interfaces aren't usually very pretty (if that matters to you) and there are certain keys they can't duplicate (such as SYMBOL SHIFT D, for instance, which might be useful in some word processing programs). You will also have to swap all the leads around every time you use a program with a different set of command keys.

Programmable interfaces are extremely versatile and (usually) very easy to set up. But they are, of course, more expensive. What you're paying for is flexibility (and some fairly sophisticated electronics!) so you must decide for yourself whether you really need the facilities this type of interface can give you.

Points to watch

Most joystick interfaces only allow you to fit one stick. A few let you fit two. Table 2.2 shows the ones which offer that bit extra. Again, as noted above, several joysticks provide an extra 'rapid fire' button, and not all interfaces can respond to this button. It's also worth checking if the interface is fitted with a 'through bus' that carries all its connections through to a socket at the back so you can add other peripherals behind it. If it is, see if it uses the full width of the Spectrum user port; if it doesn't, it won't carry *all* the connections through. Narrower connectors of this type also fit the ZX81 user port, which is smaller, so peripherals designed for the wider Spectrum port can't be fitted behind them. This is something that happens quite often with Spectrum peripherals, and it's less serious than it seems for that very reason. There are dozens of other peripherals that also use the ZX81 connector!

Talking of other peripherals, do be careful about choosing and using joystick interfaces with replacement keyboards. Some, such as the Stonechip, the Kempston and the Cambridge Intelligent Interface won't fit comfortably behind some of the new replacement

Table 2.2. Joystick interfaces.

Name	Type	No. of sticks	Through bus?	Rapid fire?	Price
AGF I	Plug	2	ZX81	Yes	£26.95
AGF II	Fixed	2	ZX81	No	£9.95
Cambridge	Prog.	1	Spectrum	Yes	£34.90
dk'tronics	Prog.	1	Spectrum	No	£22.95
Downsway	Prog	1	No	No	£25.95
E.L.R.	Plug	1	No	No	£19.75
Fox	Prog.	1	Spectrum	No	£34.95
Fuller	Fixed	1	Spectrum	No	£19.95
Jiles	Prog.	1	No	No	£24.95
	Plug	1	No	No	£19.95
Kempston	Fixed	1	No	No	£14.95
Micro-Pad	Prog.	1	No	No	£23.00
Pickard	Plug	1	No	No	£27.50
Protek	Fixed	1	No	No	£14.95
Ram Turbo	Fixed ROM	2	Spectrum	No	£22.95
Sinclair I/F 2	Fixed ROM	2	ZX Printer	No	£19.95
Stonechip	Prog.	1	No	No	£24.95
Thurnall	Prog.	2	No	No	£24.95

keyboards described in the next chapter – so think about this when you're making your choice. However, all of them will normally fit if Interface 1 is connected outside the keyboard.

One final point: quite a number of other peripherals include a joystick interface as a sort of extra bonus. These interfaces are always the 'fixed' type, and rarely allow you to fit more than one joystick – usually they use spare space on another chip, such as a speech or music chip. For the occasional games player who is more interested in music, a unit of this type could save money. You'll find full details in Chapter 6.

The detailed reviews that follow should be read in conjunction with Table 2.2, or you could miss some important points!

'Fixed' interfaces

There isn't a lot to choose between the **Datel**, the **Kempston** and the **Protek** interfaces in practice. **AGF**'s Interface II at least features a

ZX81-type through bus and it is also very cheap. When you're not using these interfaces for games, any of them will allow you to use a joystick in your own BASIC or machine code routines. Your best option is probably to look for the best price, and if a joystick is sold in the package make sure it's one that you want! The Fuller Amp Stick contains a sound amplifier and 'fixed' joystick interface for £19.95 which is certainly good value for money, and upgradable to a full sound unit. However, I think that the bigger units in this range, with other features already built in, are probably a better buy, especially as this kind of money could get you the **Sinclair Interface 2**.

Interface 2 has picked up a lot of flak in the computer press, largely because it could have been part of Interface 1 (see Chapter 5).

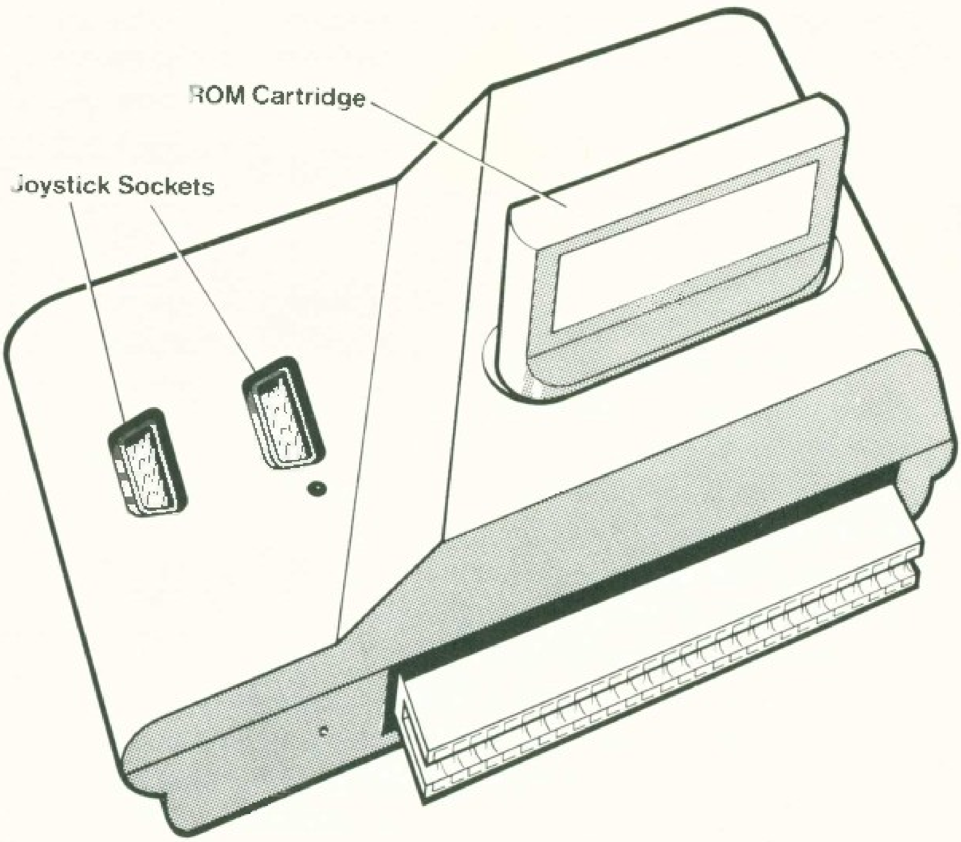

Fig. 2.6. The Sinclair Interface 2 with a ROM cartridge plugged in.

What makes it special is the socket at the top for plug-in ROM cartridges. If this sentence baffles you, take a look at Fig. 2.6, which shows the Interface 2 with a cartridge plugged in. The ROM cartridge replaces the computer's own ROM, turning even a 16K Spectrum into a perfect games machine that can actually handle games originally written for the 48K. With this system you will also be able to 'plug in' computer languages other than BASIC when (and if) the necessary cartridges are released. You can also remove the cartridge and use the interface as a conventional fixed interface in your own programs.

There are snags, though. Prolonged use may lead to the same kind of wobble I used to get on my ZX81 RAM pack, and if this crashes an arcade game just as you are about to reach the score of the century you won't be pleased. Secondly, you have to turn off the power every time you change a cartridge. Thirdly, the user port at the back will *only* support a ZX printer. Finally, £9.95 for a ROM cartridge is a bit on the pricey side, and some are still £14.95.

However, **RAM Electronics**, with their new Turbo interface, are challenging Sinclair on their own ground. Though more expensive, the Turbo doesn't require you to switch off the computer when changing ROM cartridges, and includes a full-width through bus. It had not been released when this book went to press, so check it out if you're interested.

'Plug' interfaces

The two joystick sockets on the **AGF** control the same set of keys – but at least each player can use his own favourite stick! Connections are made by fitting colour-coded leads (for up, down, left, right and fire) to wires labelled D (for Down) 0 to 4 and A (for Across) 8 to 15. You work out the position of the key you want to copy (for instance the 5 key so often used for 'move left' is D4, A5) and connect the appropriate leads to those two wires (in this case the two yellow leads). And that's all there is to it! Once the interface is set up and working, you can make a note of the arrangements on one of the printed cards supplied and store it with your program cassette. Also supplied is a rather useful 'Video Graffiti' program that allows you to draw, correct, and SAVE a complete screen using the joystick alone.

East London Robotics, makers of the Trickstick, supply a bare-board joystick interface that uses metal clips to make connections between tiny pins. The set-up is a little fiddly and confusing at first (small fingers and/or pliers help) but it works. There's a connection for Tricksticks at the back. Another bare-board interface from **Jiles** uses a small plug to shift between three different modes. With the plug connected one way the interface will suit Psion and Quicksilva software. Connected another way it makes the joystick duplicate the cursor keys.

The **Pickard Controller** is a black box filled with sockets, one for each key on the keyboard, and fitted with five plugs, one for each direction and one for FIRE. You simply plug them into the socket that matches the appropriate key – nice and simple. The box is fairly large, as you might imagine, so it actually blocks off the power socket. This is no problem, however; a plug is provided from the box itself, and you connect the power lead to a socket on the interface instead (though if you're connecting it to Interface 1 you can forget about this). By connecting the FIRE lead to the symbol shift socket you can even reach symbol shifted keys with this interface. The finish is rather crude – the plug labels on my sample were printed on a dot-matrix printer and stuck on with adhesive tape! However, this is a cheap, versatile, and ingenious interface.

Programmable interfaces

The **Cambridge Intelligent Interface** arrives complete with an excellent switch-type joystick (see above) and a software cassette. To use it, you run a library loading program on the computer, which gives you a simple choice of stored programs. You can add or delete programs as you wish, and SAVE the modified library to tape when you've finished. Once you have selected a game the loading program is deleted from the computer's memory, but the interface itself remains programmed for the game you are now ready to load.

Loading the library program takes quite a while, especially if the version you are using includes the first screen display. However, the tape supplied also contains a 'Quickload' routine that can rapidly program the interface for any given game, and the library saving routine includes an option for saving to Microdrive, which should speed things up even more. Surprisingly enough, you can't use this interface to simulate symbols or functions reached using the SYMBOL SHIFT key (not unless you use the fire button to simulate SYMBOL SHIFT). However, used carefully and with discipline, it's a worthwhile investment.

The **Fox** is an enterprising and effective interface that could save you a lot of time – but be careful how you use it, especially the first time, or you may have to write a rather embarrassing letter. Like the Cambridge interface, the Fox stores a library of instructions on board. The difference is that the library stays there when you switch off; it's permanently resident, and independent of any power supply from the computer. The secret is a trickle-charge battery that recharges itself from the computer every time you connect the interface and switch it on. If you're worried about the battery running down, don't be – it would take about a year, and a unit as versatile as this is going to be used a lot more than once a year. To keep it fully topped-up, it needs to be connected for about 7 hours a week, but it could probably manage with less. The nice thing about this interface is that there's no loading time to wait for; the interface is ready when you are. The tricky thing is that you can accidentally scrub the resident library by doing the wrong thing at the wrong time.

In this case, the wrong thing to do is to have the interface switched on (up) when you turn on the power. Fox take great care to prevent you from doing this, but it's not the end of the world if you do – you'll just have to write that letter and ask for a taped copy of the program. If that doesn't appeal, read the instructions very carefully, and then save the interface program to tape or Microdrive immediately, as suggested. This is also useful if the program is wiped out by an unavoidable accident – as happened to me just after I'd made the copy tape!

When you *do* switch on the interface you'll get a screen display showing up to sixteen program titles and key configurations in the on-board memory. You can reprogram this at any time, using single-key entries only (i.e. no shifted or symbol shifted keys, though again the shift keys themselves can be duplicated by the fire button). For advanced users, the Fox interface offers some exciting prospects. It is possible to use it for storing machine code routines, as a 'pseudo-ROM', effectively adding another dimension to the machine. This is a useful addition to your Spectrum armoury.

Another interface from **Jiles Electronics** sets out to give the big names some competition. To use it, you set it to 'program' mode using a simple switch on the top, then set up the keys you want to copy in precisely the same way as you would on the Stonechip (see below). Unlike the Stonechip, however, this interface only has to be programmed with five keys or key combinations (for up, down, left, right and fire). Although this theoretically means that fewer facilities can be programmed for operation by the joystick, in practice it means fast, simple programming and efficient operation. The switch can also be used to wipe the computer's memory without losing track of the joystick programming – a very useful feature. You can also change the programming while a game is loaded. The Jiles will copy CAPS and SYMBOL SHIFTed keys, but not EXTended mode keywords such as SQR and ABS. My only (minor) gripe is about the slightly wobbly connection, which makes it fairly easy to scrub out the computer's memory in true ZX81 fashion; but I have only seen the prototype, and I'm sure Jiles will not allow a small design fault like this to spoil such a quality product. This is an excellent interface, and certainly one of the best buys in this category. The **Downsway** operates in a similar way but lacks the top switch.

The **Stonechip** is programmed by switching it to PROGRAM, pressing the key (or SYMBOL SHIFTed key) combination you want to duplicate, pushing the joystick and/or fire button in the appropriate direction, releasing it, and then releasing the key. It will happily accept combinations of cursor keys such as ^ with > or <. The only snag with this is that sometimes you feel you need three hands, and you must also program each individual combination of functions. If you want to use the fire button, for instance, that could mean up to seventeen separate programming operations. Switch to PLAY and you can test out the programming. If you're reasonably patient, this is simple enough to do, though sometimes you have to try more than once. My only gripe is that all your painfully-entered programming gets scrubbed when you switch off the computer. So, if patience is not your virtue, choose another interface! The newer **Micro-Pad** and **dk'tronics** interfaces operate in much the same way, at a slightly lower price.

The **Thurnall** is a definite one-off – I've seen nothing else quite like it. To change the keys it duplicates you can alter any or all of eight tiny switches inset in a panel on one side. Problem one was reaching and changing the switches; I eventually used a screwdriver, as even the smallest fingers would be no good for this job. Problem two was working out how to configure this interface in any given situation – it really isn't easy to figure out which switches need to be set! The idea is good, but I can't rate this one very high.

Using your joystick

If you're going to use a joystick in your own programs then it's important to understand how the *keyboard* works, which should make a useful introduction to the next chapter! The Spectrum keyboard, as you will see in Fig. 3.1 (Chapter 3) divides each horizontal row of keys into two half-rows. Each of these can be 'read' using an IN command, as follows:

CAPS SHIFT to V	PRINT IN 65278
A to G	PRINT IN 65022
Q to T	PRINT IN 64510
1 to 5	PRINT IN 63486
6 to 0	PRINT IN 61438
Y to P	PRINT IN 57342
H to ENTER	PRINT IN 49150
B to SPACE	PRINT IN 32766

That probably looks fairly baffling, so if you have a plug or programmable interface, or a fixed interface that can duplicate the Spectrum cursor keys, connect it up and try the following program:

```
10 REM Joystick
20 PRINT AT 12,0; IN 61438;" Half row 6 to 0"
30 PRINT AT 14,0; IN 63486;" Half row 1 to 5"
40 GO TO 20
```

Now try moving your joystick and watch the numbers change. Press the keyboard keys in the appropriate half rows and you should be able to match these numbers. Notice also that you can actually press more than one key, and get a usable number. For instance, if you press keys 7 and 8 together you should get the number 243. These two keys are also the cursor keys for UP and RIGHT, so you can use the reading 243 in your programs to send something diagonally upwards to the right. If you want to see how this works in practice, take a look at the joystick subroutine at line 9300 in the Spectrum Adventure (Appendix 1). But, for the moment, I'd like to concentrate on what the joystick and its interface are simulating – that much-abused and totally necessary workhorse, the keyboard.

Chapter Three
Keyboards

If you are happy with the keyboard on your Spectrum, fine; skip this chapter and go on to something that interests you. But just before you go, think about how the Spectrum keyboard is made. The earlier ZX80 and ZX81 computers both used 'touch-sensitive' keyboards that were not, in fact, all that sensitive. The only way to be sure you had actually entered a character was to look at the screen and see if it was there! The Spectrum keyboard, believe it or not, is only a little more sophisticated: underneath that cosmetic exterior there beats the heart of a ZX81! The 'keys' are merely lugs on a single rubber sheet sandwiched between the metal cover and the keyboard circuitry underneath (see Fig. 3.1). When you press down on one of these lugs, two wires are pushed into contact with each other.

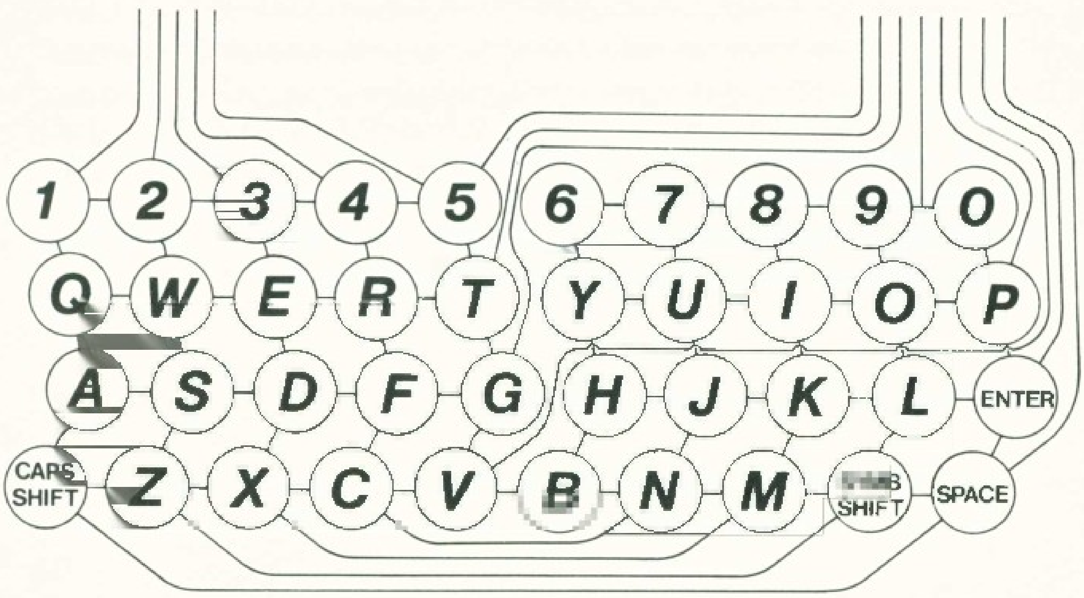

Fig. 3.1. The *real* Spectrum keyboard! Notice that five wires go to the ribbon cable on the left, and eight to the one on the right.

Depending on which connections have been made in this way, a code is generated and sent to the input port for the keyboard. This is hardly the way to make a professional-quality keyboard, as Sinclair themselves would be the first to admit – in fact the keyboard is one of the items that helps keep down the price of the Spectrum.

Even so, the Spectrum keyboard can be used for word processing. It's slow and not particularly comfortable, but the Introduction and Chapter 1 of this book were written in this way. However, in due course you may, like me, come across a more serious problem. Round about the end of Chapter 1 my beloved Spectrum suddenly gave up with typing the letters m, n, and b. It wasn't too keen on SYMBOL SHIFT, either. Notice that all these keys are on one half-row. When the first of the replacement keyboards arrived, I connected it up and found that the computer's main circuitry was perfectly OK. The trouble was purely and simply the keyboard.

If you are quite sure this could never happen to you, then don't be. For one thing the half-row from B to SYMBOL SHIFT on my Issue 2 is right on top of the heat-sink, and obviously long and hard use of the computer will have an effect on the keyboard connections immediately above. Within a week of my keyboard going down I had heard of three other Spectrum keyboards going down in exactly the same way and on the same row. The villain of the piece does indeed seem to be the heat sink. Its near presence causes slight deformation of the keyboard, and the contacts become dirty until they eventually fail to operate. It's a fault that can be put right, but it will cost you £30 and leave you without your computer for at least 2–3 weeks. For that you could buy one of the cheaper replacement keyboards!

The Spectrum keyboard is cheap, and it's perfectly adequate for an inexperienced beginner. But it will wear out, and if you're becoming a more serious user you may very well want something better anyway. So what should you be looking for?

Looking for the perfect keyboard

Everyone's idea of the perfect keyboard is going to be a little different – it depends what you want to use it for. The main uses for a replacement keyboard are:

- Word processing
- Entering data (figures or words)

- Entering programs
- Playing games

For *word processing* – writing books like this one, for instance – the keyboard should be as much like a typewriter as possible. You are looking for:

- 'Dished' key tops that will catch and hold your finger to stop it slipping.
- A 'raked' keyboard, which presents the keys to you in stepped ranks rather than all parallel to each other.
- Sprung keys that move when you touch them (so that you're sure they have responded) and are not too tiring to use.
- A proper space bar (vital).
- An extra CAPS SHIFT key on the right-hand side (vital).
- Extra 'single-entry' keys that will let you enter punctuation (commas, full stops, colons, semicolons, etc.) without pressing another key such as SYMBOL SHIFT first.
- A single-entry DELETE key. Most people use this key more than any other!
- Full compatibility with Microdrive. You're going to need the Microdrive for any serious word processing system unless you opt for disk drive instead.

For *entering data* you really need all the features mentioned above, plus a couple more:

- A numeric keypad, like the keypad on a calculator, is invaluable for entering long strings of numbers.
- A separate full-stop key is vital for adding decimal points, and should ideally be on the numeric keypad.

If your main use of the keyboard is *entering programs*, then you can probably do without some of the extra punctuation keys and you may even manage without the space bar. However, useful extra features might include:

- Single-key entry to extended mode (so you don't have to press CAPS SHIFT and SYMBOL SHIFT together just to reach the red and green keywords).
- Very clear key markings (so you can find rarely-used commands as easily as possible).
- Single-entry cursor keys (so you don't have to press CAPS SHIFT first).

- Easy connection with Interface 1 and with the full range of add-on units for the Spectrum.

If you're *playing games* you're usually much better off with a joystick! However, if you are an occasional games player with no wish to buy a joystick then look out for keyboards with:

- A numeric keypad (duplicating the cursor keys)

or

- Single-entry cursor keys

At the moment these are the Fuller FDS, the dk'tronics, the AMS, the Transform, the Saga Crusader and the Saga Emperor (though on the Emperor, AMS and the dk'tronics you'll have to hold down CAPS SHIFT as well).

The choice is yours

The rest of this chapter will tell you how the keyboards currently available shape up to those specifications (for a quick summary see Table 3.1). None of them actually has all the extra features mentioned above, so you'll need to decide which of them are most important to you. The reviews are divided into two sections according to price and the facilities offered.

Table 3.1. Keyboard features and prices.

Model	Extra keys (s = single entry)	Interface 1 fitting	Numeric keypad?	Price
AMS	CAPS SHIFT CAPS LOCK	Good. Interface 1 outside case.	Yes	£49.95
dk'tronics	CAPS SHIFT SYMBOL SHIFT SPACE BAR	Excellent. Interface 1 inside case.	Yes	£45.00
Fuller FDS	DELETE (s) EXTENDED MODE (s) SYMBOL SHIFT CAPS SHIFT 4 cursor keys (s) Space bar Comma (s) Full-stop (s)	With some alterations or with Interface 1 outside case.	No	£49.95

Fuller FD42	None	Poor. Interface 1 outside case.	No	£29.95
Maplin	DELETE (s) EXTENDED MODE (s) SYMBOL SHIFT CAPS SHIFT Split space bar	Interface 1 connects to an external PCB.	No	£44.95 (built) £35.00 (kit)
Ricoll	CAPS SHIFT Space bar	Poor. Interface 1 outside case.	No	£37.95
Saga Crusader	DELETE (s) EXTENDED MODE (s) SYMBOL SHIFT CAPS SHIFT GRAPHICS (s) CAPS LOCK (s) 4 cursor keys (s) Space bar Plus single entry keys for: > < $ / * = - + # ; : , . 7 0 6	Good. Interface 1 outside case.	No	£59.95
Saga Emperor	DELETE SYMBOL SHIFT CAPS SHIFT GRAPHICS CAPS LOCK Space bar Plus extra keys for: > < $ / * = - + # ; : , . 7 0 6 LOAD RUN LIST POKE	Good. Interface 1 outside case.	No	£49.95 (approx.)
Stonechip	DELETE EXTENDED MODE (s) Space bar RESET Tape SAVE/LOAD switch and BEEP amplifier	Poor. Interface 1 outside case.	No	£60.00
Transform	DELETE (s) EXTENDED MODE (s) SYMBOL SHIFT CAPS SHIFT EDIT (s) Space bar Comma (s) Full-stop (s) Colon (s) Semicolon (s)	Excellent. Interface 1 inside case.	Yes	£69.95

Paying your money . . .

The design specifications for the **Saga** range put most of their rivals to shame. Both the **Saga Crusader** (the mid-range model) and the **Saga Emperor** (the cheapest model) provide an astonishing range of extra keys, as the table shows, as well as the essential space bar and extra CAPS SHIFT keys. A special feature of the Crusader is its tally of single-entry graphics symbols and cursor commands. This makes the Microdrive commands especially easy to enter, and the Crusader is perfect for word processing. The casing is slim and flat – very attractive – and Interface 1 fits neatly 'outside and underneath' giving useful tilt. The Saga Emperor dispenses with most of the Crusader's single-entry keys, but the shift keys you need are placed in very convenient groups so you're not losing as much as you might think. The low price puts this range firmly at the top of my personal list, and Saga are also promising a new de luxe model, probably with a numeric keypad, in the near future. My one grouse is the stick-on key legends, which do not wear well. You can sometimes help preserve them by applying a coat of clear varnish, but I do hope the manufacturers can do something about this before full production starts. Here are my ratings for both Saga models (giving marks out of ten):

Emperor

For word processing:	9	
For data entry:	9	
For programming:	9	

Crusader

For word processing:	10	(all real requirements met)
For data entry:	9	(no separate numeric keypad)
For programming:	10	(superb provision for the new Interface 1 commands)

The **dk'tronics** keyboard also suffers from stick-on key legends, and one or two, particularly the $ sign, are a bit hard to spot. It's especially difficult to fit the stickers that identify the colours on the front of the number keys.

Dk'tronics supply twelve extra keys grouped in a block to the right of the main keyboard. These duplicate the ten number keys (and therefore the four cursor keys and all the colour keys) plus the

'APS SHIFT key and the SYMBOL SHIFT keys. As CAPS .OCK is duplicated on the 2 key immediately above, and the)ELETE key is on the duplicate 0 next door, this is a powerful combination – and one, incidentally, that even Fuller didn't think of (see below). Having CAPS SHIFT and SYMBOL SHIFT side by side obviously makes it far easier to reach extended mode and get to the red and green keywords. Provision for Microdrive is excellent: the casing design leaves plenty of room to fit Interface I inside, and a special PCB connects it to the ribbon cable for the drive units, which would otherwise be too short. Each hole at the back is neatly cut and labelled, ready for your plugs. The absence of single-entry keys is a pity, but this keyboard is adequate for word processing, and for those who use their computer mainly for programming and number work it is probably an ideal choice. Here are my ratings, out of ten:

---

dk'tronics

For word processing:	7	(no single-entry cursor keys)
For data entry:	8	(numeric keypad but no single-entry decimal point)
For programming:	9	(extra functions easily available from the numeric keypad; Microdrive can be fitted internally as standard)

The design of the **Fuller FDS** keyboard is excellent, with one glaring exception. Table 3.1 shows the many extra keys provided, and Fig. 3.2 shows how they're laid out, but two deserve special mention. On the second row up, between the L and ENTER keys, are two red keys labelled f1 and f2. F1 takes you directly into

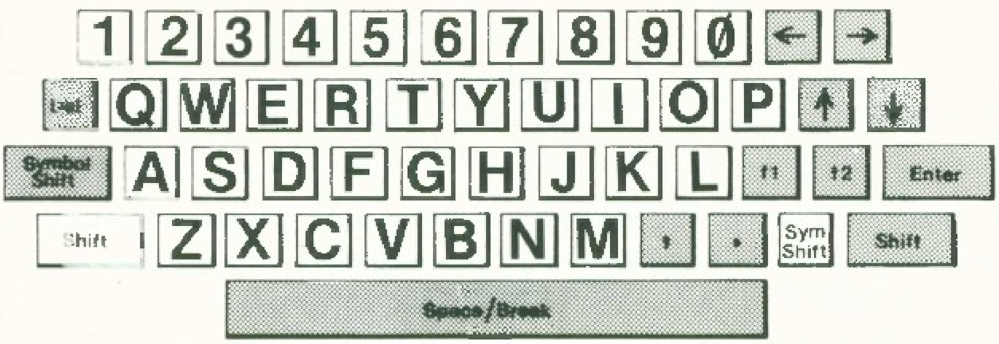

Fig. 3.2. Layout of the Fuller FDS keyboard – far closer to what a typist would expect, with some very useful extra keys (shaded).

extended mode, and as long as you hit the next key reasonably quickly you will get the keyword printed in red below the equivalent key on the Spectrum. The only problem with this is a 'delay effect': typing the Microdrive command CAT 1, for instance, by hitting f1, 9, 1 very often produces CAT! The f2 key, similarly, will give you the keyword printed in green above the equivalent Spectrum key.

These two keys alone make the Fuller worth very serious consideration: the others may or may not be important to you, but once you get used to them you will find it very hard to manage without them. The legends printed on the keys look nice, but do not wear well – and the colour labels are missing from the number keys on the top row.

The Fuller's provision for Microdrive is surprisingly poor. You can certainly fit Interface 1 into the case – for that matter you can fit the power supply inside the case as well – but the arrangements for fixing the boards are very crude, and you're invited to cut the necessary holes for extra screws and connections yourself! However, Fuller tell me *they* will make the necessary alterations on request, so if you want to fit Interface 1, ask! If you are chary about opening up your Spectrum, Fuller also offer the 'cop-out' option of leaving the computer in its existing case and connecting it to the keyboard with a cable from the user port at the back. At £10 this is an expensive cop-out, and you might do better to buy the necessary cabling and connectors more cheaply elsewhere, or even fit Interface 1 outside the keyboard. It isn't neat, but it works, and it gives a little bit of extra tilt that I find more comfortable.

The 'touch' – the pressure needed to make a proper key contact – seems to vary from board to board. Try to get one that suits you.

Fuller FDS

For word processing:	9 (comfortable responsive keyboard with useful extra keys)
For data entry:	8 (no numeric keypad)
For programming:	9 (useful extra keys)

The 'Lo-Profile' keyboard from **Advanced Memory Systems** is attractively styled in a slimline case complete with numeric keypad that includes a spare CAPS SHIFT and a full-stop. However, apart from the space bar and an enlarged ENTER key, that's it. The numeric keypad omits to mark the extra functions also carried on the duplicate keys – it would almost pay you to mark them yourself.

Interface I fits outside, but fits well – the only problem is the ribbon cable for the drives, which is too short to clear the keyboard. The manufacturers plan to supply longer ribbon cables!

Advanced Memory Systems

For word processing:	8 (no punctuation)
For data entry:	8
For programming:	8

The **Stonechip** keyboard has the advantage of saving your Sinclair guarantee (see the section below on fixing replacement keyboards). You put the cased computer inside the keyboard casing and plug an extra PCB into the user port. This PCB also has connections for the EAR and MIC sockets, the TV aerial lead, and the power lead. The far side provides duplicates of all these connections which appear at the back of the case, and this is where I found the first snag: the power socket has been moved from its usual position to the other side of the user port. In this position it fouls most of the peripherals supplied by other manufacturers – so minus quite a number of points for design. Interface 1 will fit on the back (just – connecting peripherals to this PCB seems to be rather difficult) but it sits awkwardly underneath a keyboard that really doesn't need to be tilted. You can always connect peripherals with a cable, and avoid the power lead problem altogether, but this could cost you money.

The keyboard itself is disappointing. Although the keys are responsive and reasonably well laid out, the large duplicate DELETE key at bottom right won't work unless CAPS SHIFT is pressed as well, which is precisely the sort of thing I buy a replacement keyboard to avoid. A single enlarged EXTENDED MODE key will get you to the green keywords in one stroke, but for the red ones you still need SYMBOL SHIFT, and its position on this keyboard is slightly awkward because of the extra keys around it. There is a good, solid space bar, but no duplicate CAPS SHIFT key. Two completely new RESET keys, on different parts of the keyboard, will wipe the computer's memory if pressed together – a neat and original idea. But the prize goes to Stonechip for their built-in BEEP amplifier. It operates through a small speaker mounted at the top of the keyboard, and there are also volume and tone controls and a selector switch for LOADing or SAVEing on cassette. Here are my ratings:

Stonechip

For word processing:	7 (no single-entry punctuation)
For data entry:	6
For programming:	4 (poor connection with other peripherals)

The **Transform** (see Fig. 3.3) manages to combine most of the good features of the Fuller and the dk'tronics boards, but at a price £69.95, to be exact. Provision for word processing is excellent, and there's a numeric keypad as well as the battery of additional single-entry keys shown in Table 3.1. It will even hold the ZX power unit,

Fig. 3.3. Layout of the Transform keyboard.

complete with case, inside, and you can have an on/off switch and red warning LED as easily-fitted optional extras. Interface 1, fully cased, fits snugly at the bottom of the metal keyboard casing, yet still 'inside' it – no compatibility problems here. Key legends are printed in three colours, and etched into the plastic of the keys so they should last well. For the serious user this is a keyboard worth considering, as long as you're willing to pay the asking price.

Transform

For word processing:	9 (full range of extra single-entry keys, except cursor keys)
For data entry:	9 (numeric keypad and single-entry decimal point)
For programming:	9 (all necessary extra keys are supplied, and Microdrive can easily be fitted)

... and taking your choice

So far I've concentrated on the upper range of keyboards. However, if all you want is an improved version of the Spectrum keyboard, and the features outlined above are not too important to you, there's still quite a wide choice.

The **Fuller FD42** gives you a no-frills full-travel keyboard duplicating the original Spectrum layout. The touch is pleasant, and it's neatly styled. However, there is no proper provision for Interface 1, and no extra keys at all.

Fuller FD42

For word processing:	5
For data entry:	5
For programming:	4

The **Ricoll** is the only keyboard apart from the Transform with a metal case. It looks tough and functional, but has only two extra keys, the vital space bar and the equally vital spare CAPS SHIFT key for the right hand. The keys are well-shaped and responsive, but once again they use small stick-on legends, which are not pretty and not particularly durable. Fitting the computer's printed circuit board (PCB) is easier than the instructions led me to believe but perhaps my instructions covered an earlier model. However, when separating the two halves of the Ricoll case, do lift off the base rather then sliding it off or you could damage the keyboard circuitry inside. The rear connectors give full access to the Spectrum PCB, but there is no provision for including Interface 1 or the power unit inside the case. Incidentally, Ricoll's literature claims that use of the keyboard won't invalidate your Spectrum guarantee. As you have to take the PCB out of its case to fit it, this is simply not true – the guarantee is invalidated as soon as you open the case.

Ricoll

For word processing:	6
For data entry:	5
For programming:	4

The **Maplin** keyboard plugs directly into the user port at the back of the cased Spectrum. This one is available both in kit form and ready-built, but unless you know exactly what you're doing I don't recommend the kit. Both versions 'plug into' the computer using a special adaptor; the adaptor also has a through bus, so you're not losing anything. There are extra single keys for GRAPHICS, CAPS SHIFT, CAPS LOCK and EXTended mode, plus an extra key in the kit version that can be wired for your own requirements. The kit also includes two joystick ports – a real boon. However, the assembled version doesn't include them, and the curiously small split space bar is less useful than it could be.

Maplin

For word processing:	5 (no single-entry punctuation)
For data entry:	5
For programming:	7

Fitting replacement keyboards

As we have seen, many replacement keyboards can only be fitted by taking your Spectrum, and sometimes even Interface 1, out of their cases. It's easy enough, but it will invalidate your guarantee, so don't do it lightly. *And always start by unplugging the Spectrum at the mains.*

When you have undone all the screws (you'll need a screwdriver with a 'Philips' or 'Pozidriv' type head for this) the keyboard simply lifts off to reveal the main PCB (printed circuit board) as shown in Fig. 3.4. Coming from the back of the keyboard you'll see two ribbon cables that plug into connectors on the PCB. All you have to do is tug (gently) at each ribbon cable, and it will slip out of the connector. To remove the PCB from the lower casing, just undo the small screw near the centre: you should be able to lift it out easily. Taking Interface 1 out of its case is just as easy, but be particularly careful when you are easing the edge connector out of the hole in the top of the case: there's a row of fragile-looking metal wires just behind it, and it is very easy to put damaging pressure on them. Breaking any of them will ruin the Interface: and you've just invalidated the guarantee!

In the case of the Fuller FDS you need to plug your power supply directly into the keyboard. A feeder wire takes it down to the Spectrum's normal power socket. I am told this can cause trouble

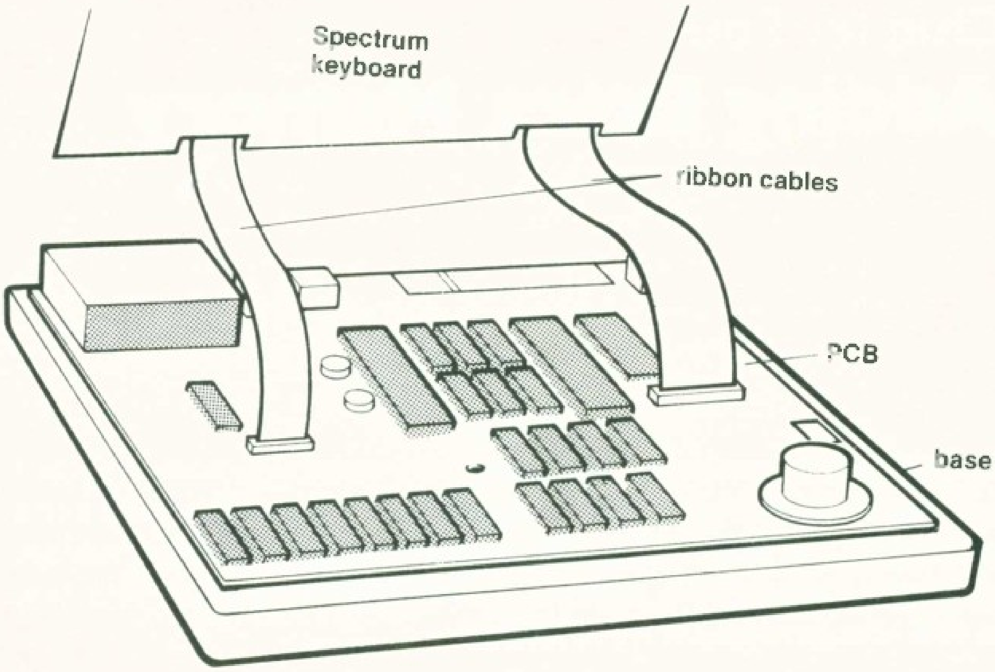

Fig. 3.4. Fitting the new ribbon cables to the Spectrum PCB.

when running some machine code programs, but I can't say I've noticed this myself. If you haven't taken your power unit out of its case and refitted it inside the Fuller case (and frankly I wouldn't recommend it) you may hit a snag at this point. You will have to find somewhere to pass the two power cables – the one from your power unit and the one from the keyboard – through the casing. No provision is actually made for this if the PCB is mounted inside the case, so again it may well be a question of drilling your own holes. Frankly I'd be inclined to pay the extra for the linking cable and leave the Spectrum in its case.

Making your choice

In the end, choosing a keyboard is a matter of personal taste, and depends very largely on what you want to use it for. I hope that these brief reviews will at least give you some idea of the ones you are most interested in. After that it's up to you. But remember you are going to live with your choice for quite some time – so don't be bullied or cajoled into buying a keyboard that you are not quite sure about.

Chapter Four

Adding Extra Memory

This chapter is purely for 16K Spectrum owners who want to get out into the wider world of 48K computing. It is possible to add extra memory to the 48K, but frankly you'll be better off with a storage system like Microdrive. However, if you want to upgrade your 16K to 48K you have four ways of doing it:

- Ask Sinclair to do it for you.
- Ask a competent dealer to do it for you.
- Buy an upgrade kit and do it yourself.
- Buy a plug-in memory unit.

Each of these options except the first and the last involves a series of steps that could well make your blood run cold. The 'upgrade' is in fact rather less simple than it sounds! The first step is to open the case. As we saw in the previous chapter, this immediately invalidates the Sinclair guarantee. Next you have to take individual microchips and fit them to the main printed circuit board (PCB for short, if you skipped Chapter 3). If you've never actually seen a microchip before, Fig. 4.1 will show you what you're up against. Most of the chips in the upgrade kit are about ¾ inch long.

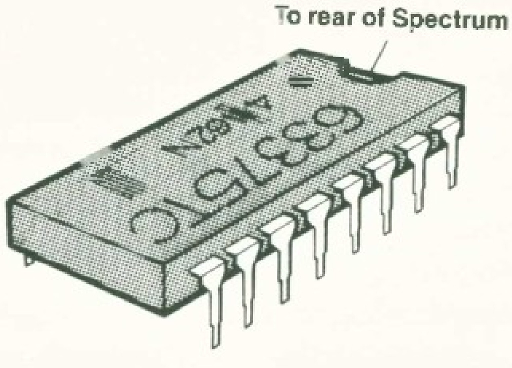

Fig. 4.1. A microchip.

Careless handling at this stage could damage one of the chips (annoying and time-wasting) or even, if you were very unlucky, the computer's own PCB (disastrous!). Assuming that all is well, however, it's still possible to insert a chip the wrong way up. At best, this means your upgrade won't work. At worst, you could permanently damage other chips on the main PCB. A good upgrade kit, of course, comes with clear instructions and detailed diagrams, but unless you are very confident that you can cope, beware of carrying out upgrades yourself. For those who want to, there are detailed instructions at the end of the chapter. All the same, this is undoubtedly the cheapest way to reach the magical 48K.

Making your choice

This section doesn't contain the detailed listings of manufacturers and products you'll find in most of the other chapters. For one thing, many Spectrum upgrade kits are sold to dealers rather than direct to the general public. For another, the only thing that's really different about upgrade kits is the quality of the instructions: most sell for around £25. And finally, there are so many electronics firms 'having a go' at this sort of thing that a detailed review of all their different instruction sheets would be out of date tomorrow. Much the same situation applies to plug-in RAM units. Prices are pretty stable for these units at about £40. They either work or they don't: if they don't, send them back and demand one that does, or your money back. Even the best hardware companies sometimes have a bad day.

Going to Sinclair

The main advantage here is peace of mind – your Sinclair guarantee remains unaffected. Snags are the price – currently around £40 – and the delay, which will be at least two to three weeks. For £40 you could actually buy a plug-in memory unit and have your 48K Spectrum instantly, so unless you're desperate to have your memory unit inside rather than outside the case, this really isn't the best option.

Going to a dealer

Snag number one is finding a *competent* dealer you know you can trust. For this it's hard to beat personal contact and personal recommendation, so talk to your local User Group. If you don't know where they are, look in the Sinclair-dedicated magazines like *ZX Computing* or *Sinclair User*, or check at your local library. If fitting the upgrade is going to mean invalidating your Sinclair guarantee, check that the dealer who does the job is offering some guarantees of his own.

Doing it yourself

To check how easy this was I looked at two kits, one from Fox Electronics and one from dk'tronics. The Fox kit arrives in a clear plastic box, with the chips bedded on polystyrene foam and the instructions folded above them. It's very well documented, and good value. The dk'tronics kit arrives with the chips already in place on a neat diagram of the main PCB – ten out of ten for presentation. Both are intended for Issue 2 machines. At the time of writing, Issue 3 16K machines are not all that common in the UK, but in any case you will find the issue number printed in white ink along the bottom of the PCB. If you have an Issue 3 16K then the upgrade kit from East London Robotics should fit it.

Getting ready

The first thing to watch out for is static electricity. It has a nasty habit of scrambling the information that's been programmed into your microchips. To avoid such disasters:

- Wear nothing on your forearms.
- Sit still at the table for a while before opening the upgrade pack.
- Don't shuffle your feet on the carpet while you're working.

Now open up the Spectrum case as described in Chapter 3. If you like, you can remove the keyboard altogether by *gently* unplugging the ribbon cables from the sockets on the main PCB. The dk'tronics kit contained dire warnings against this, but I can't see why. You have to do this to fit their keyboard! If you now look at the main PCB you should be able to see twelve empty spaces just begging to have microchips plugged into them. Fortunately eight of them, the memory chips, are exactly the same! For that reason, if nothing else, start with the memory chips.

Fitting the chips

Priority one is to get the chip the right way round. Figure 4.1 should help here – the half-moon shaped cutout or darker dot should be at the back, pointing to the Spectrum's rear connections. Some chips use both markings. The instructions in your kit will tell you which are the memory chips – they're identified by the number printed on the chip, though confusingly enough some manufacturers add extra numbers before and after the identifying code. Take your time about this.

Once you know which chip you want, check that you know where to put it. The memory chips go in the eight empty spaces near the front of the PCB (see Fig. 4.2). Pick up the first one, but be careful not to grip it so that you're squeezing the pins – they bend! Hold it with your thumb pressing the front end and your finger pressing the back end, and keep it absolutely level with the PCB as you position it over the socket. If the pins are splayed out, and won't fit in the socket, don't try to bend them with your fingers. Lay the chip on its side, with the pins resting flat on the table, keep your grip on the front and back, and push the chip *gently* until the pins are at right angles to the body of the chip itself.

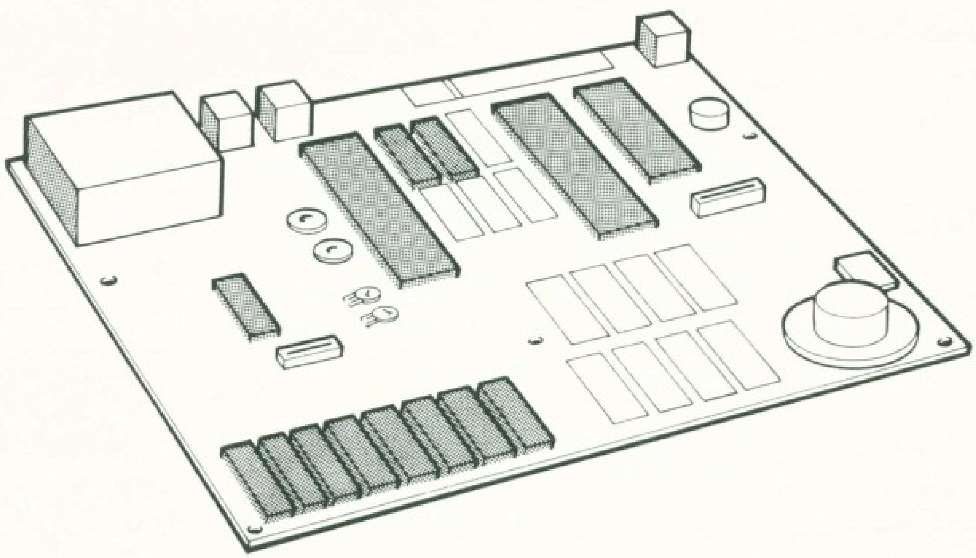

Fig. 4.2. 16K Spectrum PCB showing spaces for new chips.

When fitting the chip, use the forefinger of your other hand to do the pushing while you steady the chip with the grip described above. Push it in *evenly* – bending the pins could cause serious damage – and check that all the pins are properly located. If one of them isn't, you could easily bend it completely out of shape by pressing down on the chip. If you've made a mistake, or something goes wrong, lever the chip out gently with a screwdriver, a very small amount at a time, starting at one end and switching to the other so that the chip remains as level as possible all the time.

The remaining four chips are all different. Fit them according to your supplier's instructions, and make sure they are all in the right place and the right way round. Check all the chips this way, and check them again. At this stage it's difficult to be too careful. Some upgrade kits require you to cut wire connections on the PCB. If yours does, take great care with it! When you're satisfied that all is as it should be, replace the keyboard and close the case.

Testing your upgrade

Now comes the moment of truth. Make sure the TV is connected and switched on, then plug in the power lead at the back of the computer and switch on at the mains. If you don't get the normal black screen and copyright notice within four seconds then switch off – *fast!* Something is obviously very wrong. Open the computer and double-check your connections. If you can't find the error, then there are two nasty possibilities:

- The upgrade kit is faulty in some way.
- Your computer's PCB is faulty or damaged.

In either case you'll have to remove the upgrade chips – carefully! – and pack them up again. Send them back to the supplier for checking. Meanwhile, once your computer is back to its old 16K self, plug it in and check it out again. If the fault persists, then you'll have to get *it* repaired as well – and hard luck. If all is well on the first test, and you get a copyright notice, then try entering:

```
  10 LET memory=(PEEK 23732+256*
PEEK 23733)+1
  20 LET size=memory/1024-16
  30 PRINT size
```

Fig. 4.3. Simple memory check routine.

When you RUN this program you should see the magic figure 48 appear on screen – not an acid test that all is well, but a very good indication that it might be.

Your upgrade kit should normally include a fuller test program that tests out the available memory completely. If it doesn't, try this program:

```
  10 CLEAR 32000: LET test=0
  20 FOR m=32768 TO 65535: POKE
m,test: PRINT AT 0,0;m: NEXT m
  30 FOR m=32768 TO 65535: PRINT
 AT 0,0;m: IF PEEK m<>test THEN
  GO TO 70
  40 POKE m,255: NEXT m: IF test
 THEN  GO TO 60
  50 LET test=255: GO TO 30
  60 PRINT "Relax. Everything ch
ecks out": STOP
  70 PRINT "Sorry, There's a fau
lt at ";m
```

Fig. 4.4. Full memory check routine.

Don't sit by the screen. It will take quite a while so go and make yourself some coffee while you wait. All being well you should get the friendly message in line 60. If not, you'll have to remove the upgrade kit and send it back for checking.

Plug-in RAMpacks

This is the easy way out, though not the cheap way. Prices are typically about £40 for a plug-in pack, compared with £20–£25 for an upgrade kit. Again, you're paying for convenience and peace of mind. However, check that your plug-in has a proper through bus (all the ones I've seen had one) and check that it's a snug, secure fit in the user port or behind Interface 1. That's right, *behind*. That may sound odd, but when I tried putting one between the Interface and the computer, where I'd expect it to be, I achieved a fairly spectacular lock-up. A tight fit is important; loose-fitting add-ons will crash your computer with monotonous regularity.

Some manufacturers offer double-size RAMpacks that give your computer up to 64K of RAM. Before you go wild with excitement, I should point out that you can't get to all 64K at once – you have to use a 'switching command' which effectively shuts out one 'page' of the extra RAM and cuts in another. This is because the Spectrum, as described in Chapter 1, can only cope with 65536 addresses; it has no way of reaching addresses with a higher number. Units like these may sound attractive, but they are really for machine code programmers; it's hard to use them properly from a BASIC program. The remarkable range of add-on units from Basicare, which could theoretically give you *several thousand* K of 'paged' memory (see Chapter 11), is also for serious users – and it's rather complicated. You'll really be better off with a simple fast-access storage system like Microdrive. Talking of which, it's about time to move on to a closer look at storage systems in general, and see what's available and what is likely to suit your special needs.

Chapter Five

Adding Extra Storage Space

If you are just starting out with your Spectrum, then the idea of hundreds of kilobytes of storage space may seem irrelevant to you. After all, what would you use it for? The answer will soon become very clear – you would use it for the enormous mass of part programs, junked programs, and the odd one or two successful programs that everyone seems to generate in their first few weeks at a Spectrum keyboard. The moment you put together your first program and save it on a cassette tape your problems are beginning. In fact one of your early priorities should really be a database program like the one in Ian Sinclair's *The ZX Spectrum and How to Get the Most From It* (Granada). You may not think that you are going to need a special program to find your programs, but believe me you will.

The trouble with cassette storage is speed (or rather lack of it) and accessibility. On a long tape, the only way to get to the program you want is by wading through a good many others that you don't want. Of course, you can buy much shorter cassette tapes – and finish up swimming in them with nowhere to put the furniture or, come to that, books like this one. So what are the options?

The filing file

For Spectrum owners there are three main choices:

- Cassette tape.
- Microdrive.
- Disk drive.

Cassette tape is slow to LOAD and SAVE, and difficult to access quickly. It can also be unreliable. Normally you *must* VERIFY to make quite sure you've got a decent copy of your program.

However, this is ultra-cheap storage; it uses hardware that most people already have, and cassettes are available on any high street.

Microdrive provides rapid access to about 90K of information stored on a tiny tape cartridge. A special drive unit is needed, and Interface 1 must be fitted to your Spectrum before the unit can be used. Up to eight drive units can be chained together. Typical loading time for a BASIC program is seven seconds. The cost for a starter system including Interface 1, one drive, and two blank cartridges is about £90 if you buy direct from Sinclair.

Disk drive, the serious user's storage system, is now available for the Spectrum. It allows extremely fast access to at least 100K. However, it's expensive compared to the cost of the computer itself. Prices start at about £300, including the necessary interface and in some cases a suite of software.

One of these systems will be right for your particular needs – and for your particular pocket. But if you feel that even Microdrive is way above your budget, don't despair. Cassette software isn't going to go away, and there are a dozen manufacturers all waiting eagerly to overcome some of its more irritating faults and failings.

Coping with cassettes

There doesn't, at first sight, seem to be much that you can do about the main limitations of cassette storage – slow LOADing and SAVEing and slow access to stored programs. However, if you check out the small ads in magazines like *Sinclair User* or *ZX Computing* you'll find several people advertising fast loader programs for cassettes that can double the speed of all your cassette operations. The price you pay for this is accuracy – so you had better make sure that you VERIFY every program! – but fast loading is possible without any fancy hardware at all. How? To answer *that* question I'll need to make a small detour into technicalities.

Baud with waiting?

When you SAVE a program to tape the CPU feeds all the data it has stored about that program to the cassette output port, one byte at a time. If it did this at its own speed the result would be an enormous mass of electrical signals emerging from the port in a fraction of a second – too fast, in fact, to be successfully recorded in any intelligible way by your cassette recorder. So the flow of data has to be 'slowed down' by the computer. The speed at which data is sent out from one unit to another is known as the *baud rate* (after a Belgian telegraph engineer called Baudot), and a unit sending at 300 baud will transmit about 30 bytes in a second.

Now since the baud rate for sending and receiving data is controlled by the computer, you can actually persuade your Spectrum to double it by some cunning programming. This will halve the time needed for transfer of data to and from cassette, but it will also halve the reliability of the transfer. The rate was set at its present level for a good reason! So changing the baud rate doesn't guarantee a faster transfer. After all, if you have to do the job twice or three times it'll actually be *slower*!

Beefing up your cassette system

If you have trouble with your cassette storage, don't worry. You are not alone. Although the Spectrum is one of the most reliable micros for cassette storage, it and many others don't always work well with *every* cassette deck. The trouble is that cassette decks are really jumped-up dictation machines – they were never *designed* for data storage.

Your computer generates signals at a fairly high frequency – one, incidentally, that some very old machines have trouble coping with. New machines, on the other hand, tend to be looking for bottom notes that are hardly there, and amplify anything they can find to fill the gap. This can confuse your computer when you try to reLOAD the recording! Stereo heads can confuse it, too: sometimes they set up a pattern of interference on the very signals the computer is trying to read. The best bet is a cheap and unpretentious deck (but try it out first!) or a deck specially made for computer work (though they tend to be grossly overpriced for what they are). Even if you don't have trouble of this kind, you are pretty soon going to be very fed up with plugging and unplugging the lead from the EAR socket every time you LOAD and SAVE. But fear not – there are at least a dozen different manufacturers all falling over each other to sell you their particular solution to the tape deck blues.

The obvious place to start is with the cassette deck itself. Is it properly adjusted? If you're not sure, then Hilderbay can supply you with a useful alignment tape and manual that will allow you to check the *azimuth adjustment* on your recorder without using measuring instruments. In ordinary language that means you can find out if the record and playback head is properly aligned: if it isn't, buy the manual and read up what to do about it! In fact, the adjustment is fairly simple (see Fig. 5.1 for the general picture). Just make sure that the 'slot' at the front of the head runs precisely parallel to the tape. If this still seems a bit mysterious, you might like to consider a recorder specially 'tuned' to match ZX computers, also available from Hilderbay.

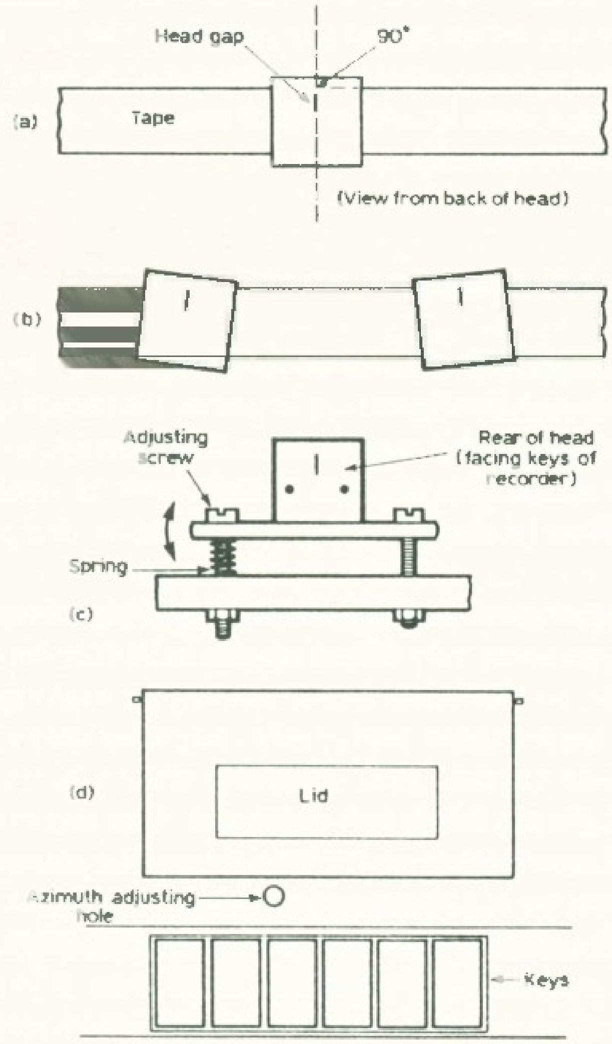

Fig. 5.1. Azimuth adjustment on an ordinary cassette deck. The narrow slit in the tape-head (a) is normally at 90° to the edge of the tape. This is the correct azimuth angle, but a surprising number of recorders have this maladjusted. Any deviation from this angle (b) causes muffled sound and poor loading. The angle can be altered by turning an adjusting screw (c) which is on the head mounting. This is often reached through a hole in the casing of the recorder (d). (Courtesy of Keith Dickson Publishing.)

Next, what about that EAR lead? **Elinca** have the obvious answer, a simple three-way switch marked SAVE, LOAD and AMP. You leave the leads connected all the time: an LED reminds you of the setting and there is a meter and filtering unit to help you get the best possible results. When you aren't using the cassette deck you switch to AMP and use it as a BEEP amplifier – nice and simple and about £14.95. The CO-DER unit from **Jiles Electronics** has an amplifier and a filter to improve recordings, and a monitor that allows the computer to check that the program is being correctly recorded. If you have ever junked a cassette as unLOADable, you may regret it now! Another option is to use a sound and joystick interface such as the **Fuller Box**, which also has a built-in self-switching interface for cassette LOADing and SAVEing, and enhances incoming signals from the deck before they reach the computer. You'll find more about the Box and its capabilities in Chapter 6.

The **Stonechip** BEEP amplifier also includes a tape loading switch and a CUE button so you can use its built-in speaker as a microphone, and record program titles or other comments onto tape before downloading material from the computer. The sound it makes is very good, but the connections on mine were a bit loose and crashed several of my programs – you have to route the 9 volt supply from the power unit through the amplifier.

Finally, don't forget the range of power supply add-ons from **Kelwood**. Many of the variants in this range include tape LOAD and SAVE switches, as well as on/off switches for the entire system (another useful extra), and a neat storage place for awkward things like cabling, power units, and multiway mains adaptors. The power base range, which sits underneath the computer, doesn't (at the moment!) make provision for Interface 1. However, the larger mains unit, which arrives complete with a three-way mains adaptor, tape LOAD/SAVE switch, mains switch, speaker with volume control, and a compartment for the Sinclair power unit, is obviously OK for any system.

Network Computer Systems offer a 'multisave' unit that will allow you to save programs to two tape decks simultaneously. **Ness Microsystems** of Inverness can supply a unit that actually controls up to two tape decks with BEEP commands! SAVE and LOAD switching is automatic (lovely!). The unit also includes an amplifier, and sells for £18.45 in kit form and £21.45 ready-built.

The more expensive units of this type are beginning to move towards a more useful kind of storage unit – one that is interactive with the computer, controlled by lines within the program itself.

Wouldn't it be nice if your Spectrum could search an entire cassette tape 15 metres long in eight seconds, pick out the program, routine, or data that it needed with unerring precision, and LOAD it almost instantaneously – and all from your pre-programmed instructions? No need to just dream about it – it's possible, thanks to Sinclair's new *ZX Microdrive* storage system.

Memory expansion on a tape

The heart of the **Sinclair ZX Microdrive** system is a tiny tape cartridge 30 × 42 mm square and 5 mm thick. You could pack about ten of them into the average cassette box. Inside is a good grade of videotape cut into a strip half the width of cassette tape (1.9 mm to be exact) and joined into a continuous loop 15 metres long. Because the loop is continuous, and because it moves past the 'playback' head of the Microdrive unit at high speed, the Microdrive system is almost like an extra chunk of RAM. It allows you to store the memory-eating sections of a program – data, arrays, and even entire routines – in a readily accessible form that doesn't tie up the computer itself.

But what amazes anyone used to cassette loading is its speed of operation. A typical BASIC program will LOAD in about seven or eight seconds. A machine code program could be even faster. And a SCREEN$ will appear on screen in about four seconds. Given that you can store up to eighteen of them on one cartridge, and at least three in resident memory, you could produce animation sequences that way.

If £90 still seems a lot to pay for fast LOADing and SAVEing, remember that Interface 1 does a lot more than connect the Spectrum to the Microdrive unit. First, it contains a built-in RS232 interface (see Chapter 8) which allows you to hook up a printer directly. Secondly, it contains a networking interface (see Chapter 9 – for the moment it's enough to say that a pair of networked Spectrums can exchange massive programs at very high speed).

The Microdrive in close-up

I could easily spend the rest of this book talking about Microdrive, but there are plenty of other books, such as Ian Sinclair's *Make the Most of Your ZX Microdrive* (Granada), that cover the subject in far more detail than I have space for. This section is just a simple introduction. All the same, I hope it gives you a few ideas.

The assembly for the Microdrive unit is as shown in Fig. 5.2, with Interface 1 mounted behind the Spectrum and the Microdrive unit attached to one side of Interface 1 by a ribbon cable. Do be careful how you connect this – it only fits one way! On the other side of the drive unit is a second socket that takes the special connector supplied in the Microdrive pack. This connector will link the unit to a second Microdrive.

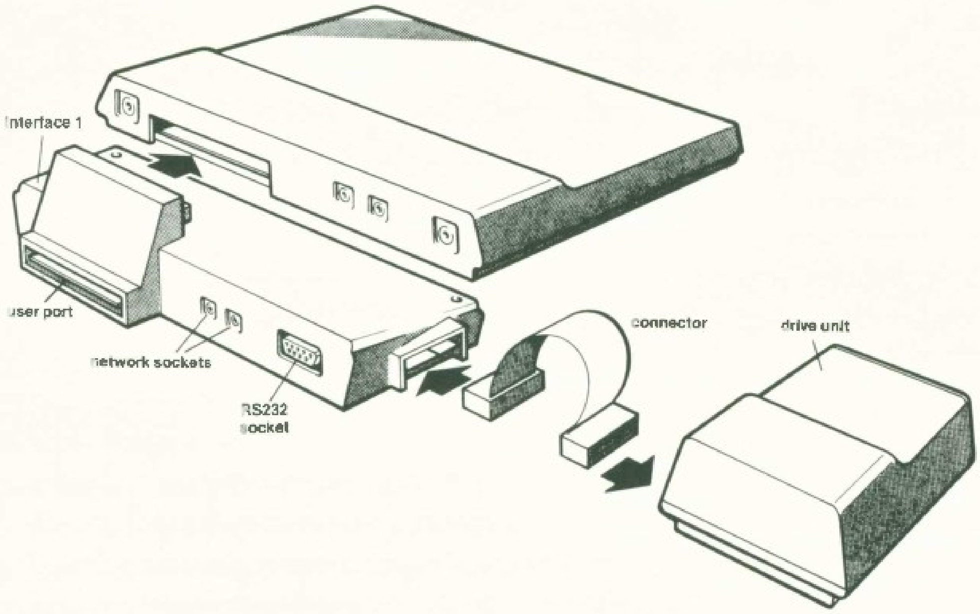

Fig. 5.2. Fitting Interface 1 and Microdrive.

A demonstration cartridge is supplied with the Microdrive. Switch on the system (this is important!), take it out of its case and slot it into the drive (Fig. 5.3). The plastic gripper at the bottom is *supposed* to come off, so don't worry if it does. Now key in RUN (the single keyword) and ENTER. The warning light on the Microdrive will come on, the drive will whirr entertainingly for a few seconds, and a menu will appear on screen. If nothing happens, or an error message appears, then you haven't loaded the demonstration cartridge. The demonstration cartridge and manual are themselves a good introduction to the Microdrive, but it does help to know something about the system before you buy, especially as a whole new set of BASIC commands are available through Interface 1.

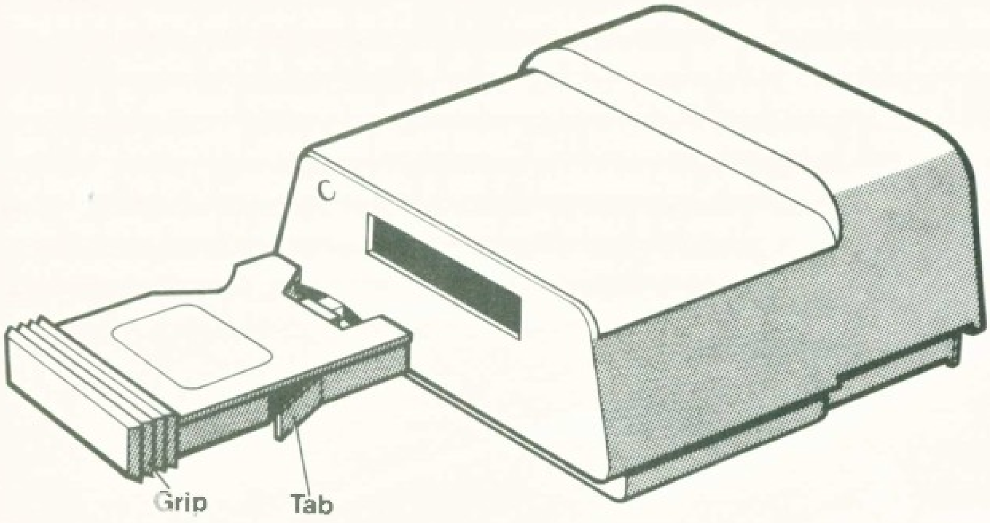

Fig. 5.3. Inserting a cartridge into the drive unit. The tab can be broken off rather like the tab on a cassette tape to prevent you from ERASEing a file accidentally.

Speak gently to your Microdrive ...

If you've ever wondered what all those extra commands on the top row were for, this is when you find out. You'll also need to get to grips with some new ideas about your Spectrum and the way it works. But let's start with that top row, where the mystery keys, in order, are OPEN#, CLOSE#, MOVE, ERASE, CAT and FORMAT. For reasons that will become clear, I'll deal with these in reverse order.

When you buy a new blank cartridge (other than the demonstration cartridge) you are buying 15 m of blank tape. In order to use this tape, the Microdrive has to 'mark up' the tape into *sectors*, and this is done by keying in:

```
10 FORMAT "m";1;"title"
```

'Title', of course, can be anything you want to call that particular cartridge – 'Games', or 'Addresses', or in my case 'Add-ons'! Each sector of a formatted tape can hold up to 512 bytes (in other words, half a kilobyte), and anything you store on a cartridge is stored in half-kilobyte chunks – even if the last section of what you're copying takes up less space than that. But the FORMAT command also 'checks out' the tape in the cartridge – and if a sector that is damaged or unusable is found, then it is effectively marked as unusable. One such section is the splice where the two ends of the tape loop are joined; others may be caused by careless handling or other damage. The speed of the tape past the heads is remarkably high (15 m in seven seconds must mean at least 2 m a second!) and sometimes it can buckle or crease – after all, it is only 1.9 mm wide! If reading this doesn't convince you that even Microdrive cartridges need back-up copies, then it should. Microdrive gives you a remarkably fast and effective storage medium, but it's not infallible.

Now that your tape is formatted, you're ready to store something on it. The BASIC commands for SAVEing to Microdrive or LOADing from it are very similar to the ones you use for ordinary cassette storage. However, each of them starts out with a variant on that jumble of letters, numbers and punctuation at the start of the FORMATting command. To SAVE a program, for instance, you would type in:

```
10 SAVE *"m";1;"program"
```

To LOAD a program from Microdrive you would enter:

```
10 LOAD *"m";1;"program"
```

To VERIFY, and to LOAD or SAVE machine code, screen strings (SCREEN$) or data arrays you use the same format, with the extra details also used for cassette storage. So far these statements don't look very different from the ones you're used to, but there are two more you will not have come across before: CAT and ERASE. For instance

```
10 CAT 1
```

will give you a list of up to 50 file titles that are stored on the cartridge in the drive at that moment – it even puts them in alphabetical order for you. No more wondering where you stored that vital set of data! ERASE is used rather like a storage command, in the form

```
10 ERASE "m";1;"program"
```

for instance. However, you don't have to add anything extra for machine code or data files. If you've stored a screen string called 'Fred' and you don't think Fred will be flattered by it, then ERASE "m";1;"Fred" will remove it for you.

Channelling the stream

I've left the three most difficult commands to last, and you won't be surprised to learn that understanding them means understanding the 'jumble of letters, numbers and punctuation' at the beginning of the other commands.

"M" in double quotes refers to the Microdrive. If you don't include it in commands like LOAD*"m";1;"program" then the computer simply won't accept your command. There are other letters that can also be used here, of which more in a moment. The number 1 is equally important – it tells the computer which drive unit you want to use. However, anything in double quotes in a command like this will be interpreted by your Spectrum as referring to a *channel*. Even if you don't have a Microdrive at the moment, there are already three channels working in your system: channel "k" (the keyboard channel), channel "s" (the screen channel), and channel "p" (the channel for the ZX Printer – which is present even if you don't happen to be using it).

On their own, these channels are useless. Until they can be connected there is no way for information to pass from, say, the keyboard to the screen. Normally the on-board programming will handle routine operations like this, but those without Microdrive might like to try the following experiment. Enter this short program:

```
  10 PRINT "Channel K to channel
 S"
  20 PRINT #2;"Channel K to chan
nel S"
```

and RUN it. You will find that the computer treats both lines in exactly the same way, by printing the message to the screen. So what exactly is '#2'? In fact it's a *stream*, and the task of a stream is to form a link between channels. If that sounds a bit wet, try the next program

```
  30 PRINT #1;"Channel K to chan
nel S"
  40 PAUSE 0
```

and your message will appear in that previously unreachable space at the bottom of the screen. Line 40 is necessary to stop the computer's 'OK 30:1' message from blotting out what you've just printed. Now try this one:

```
  50 PRINT #1;"Do you want to pr
int this line  again (Y/N)?"
  60 PAUSE 0
  70 IF INKEY$="Y" OR INKEY$="y"
 THEN  GO TO 50
  80 PRINT #0;"All right, be lik
e that"
  90 PAUSE 0
```

If you keep pressing the Y key you will eventually get an error message saying 'Out of screen 50:1', but you get the idea – here is a way of programming a menu without using the INPUT command. Used carefully, it can get you into your programs quickly and efficiently. Stream 3, incidentally, connects with the ZX Printer, so it will only work if you have the printer (or one like the Alphacom, which uses the same commands) actually connected. Using streams in your own programs can be very helpful – it means, for instance, that a simple number formula can replace a more complex choice between PRINT, LPRINT and INPUT. Try:

```
 100 FOR j=0 TO 3
 110 PRINT #j;"They seek me here
, they seek me there"
 120 NEXT j
 130 PAUSE 0
```

You should finish up with the message at the top of your screen, two at the bottom, and another on your ZX or ZX-compatible printer.

Channels, streams, and Interface 1

The moment you fit Interface 1 to your Spectrum, things begin to get even more interesting. Streams 0 to 3 are already 'spoken for' by the computer, but with Interface 1 you get at least two more channels ("m", a file on a Microdrive cartridge, and "n", another computer on a network) and *twelve* more streams, which are not spoken for at all. The value of all this depends pretty much on your own imagination. If you feel like some fun, try:

```
  10 OPEN #4,"S"
  20 IF INKEY$<>"" THEN  PAUSE 3
0: PRINT #4;INKEY$;
  30 GO TO 20
```

This has the curious effect of letting you type directly onto the screen. DELETE and CAPS LOCK won't work, but most other keys will, including the cursor keys (though you can't actually see where the cursor is until you start typing!). What you have actually done is to open a stream to the TV screen and feed codes from the keyboard directly to it.

Streams and Microdrive

If you want to try something more useful, enter the program listed in Fig. 5.4 and RUN it. When entering room names, don't use more than 20 characters or the last few letters will be chopped off in the second part of the program. Try it now, before you read any further.

```
  10 OPEN #4;"m";1;"rooms"
  20 FOR j=1 TO 4
  25 INPUT "Room number "+STR$ j
+" description?"'r$
  30 PRINT #4;r$
  40 NEXT j
  50 CLOSE #4
  60 OPEN #5;"m";1;"rooms"
  65 DIM r$(4,20)
  70 FOR j=1 TO 4
  80 INPUT #5;r$(j): PRINT r$(j)
''
  90 NEXT j
 100 CLOSE #5
```

Fig. 5.4. BASIC filing program using Microdrive.

So, what did it do? Let's take it step by step. In line 10 it linked stream 4 to a Microdrive file called "rooms". You probably heard the Microdrive whirr and then stop – actually it was looking for a file called "rooms" in case you'd already entered one. It was also setting up a clear space of 512 bytes *inside the computer itself* for storing the data coming in along stream 4. This clear space is called the *buffer*, and it's a term we'll be coming across again.

Lines 20 and 25 are straightforward enough, but what about line

30? Up to now you'll have been used to using the PRINT command to feed data to the screen – but this data isn't printed on the screen at all. Instead it's PRINTED along stream 4, and into that 512-byte buffer. And when j reaches the value 4, the stream is closed, in line 50. The command CLOSE #4 doesn't just 'switch off' the stream: it also empties the buffer – all the data in the buffer is transferred to the Microdrive cartridge. If you are PRINTing a lot of information to the buffer then this will sometimes happen during a program, before you CLOSE the stream, when the buffer has received a full 512 bytes. You'll hear the Microdrive whirr into action to deal with it.

The next part of the program reads back what you've put onto the Microdrive file, only this time it stores it in a data array. The DIMensions of this array are set in line 65. Line 60 sets up another buffer, but because the file "rooms" already exists on the Microdrive cartridge the results are rather different. This time the contents of the Microdrive file are loaded into the buffer. Incidentally, if you tried to PRINT anything to stream 5 after the OPEN # command in line 60 you'd get an error message saying 'Writing to a "read" file'. You can't add anything directly to a file stored on Microdrive. If you want to change it you'll have to move its contents into the computer's on-board memory, alter them there, and then store them back on Microdrive – and if you want to keep the original file on the same cartridge, you'll have to give your altered file a different name.

Line 80 does a familiar job in an unfamiliar way. Instead of taking INPUT from the keyboard, in the way you're probably used to, it takes it from the buffer. The effect is much the same, though, and in this case each string read by line 80 is given a number in the r$ array. Once the program has finished, and stream 5 is closed, the computer will have an array of four strings stored in RAM. To check, try entering PRINT r$(1) as a direct command, and the first entry you made should appear on screen. With a few modifications, you can use a version of this program to store room descriptions, messages, and other data for the Spectrum Adventure listed in Appendix 1.

There's one final command we need to look at. If you entered the Microdrive program for 'rooms' earlier on, just key in, as a direct command:

```
10 MOVE "m";1;"rooms" TO #2
```

and press ENTER. TO should be entered using SYMBOL SHIFTed F, of course. You should now see the contents of the 'rooms' file appear on screen – and yet any program you had in the computer remains totally unaffected. MOVE has even OPENed and CLOSEd the file for you. This is a very useful trick – you can even use it to send a data file to your ZX or ZX compatible printer by replacing #2 with #3. If you can get hold of two Microdrive units and a ZX-type printer you might like to try:

```
  10 MOVE "m";1;"rooms" TO "m";2
;"rooms": MOVE "m";2;"rooms" TO
 #3
```

The results are quite entertaining; and the whole thing happens without a screen display, and without affecting any program that happens to be in the computer at the time. Notice that files on different drives can have the same name. You can actually MOVE files to a different place on the same cartridge, but in that case you'll have to give the copy file a new name to avoid confusing your poor innocent little Microdrive. This will only work with files that have been PRINTed to Microdrive using the OPEN # and CLOSE # commands, but you can always convert ordinary data files to this format. If you had a file called 'load' containing 35 strings of characters you could enter a program like:

```
  10 LOAD *"m";1;"load" DATA l$(
)
  20 OPEN #4;"m";1;"data"
  30 FOR j=1 TO 35
  40 PRINT #4;l$(j)
  50 NEXT j
  60 CLOSE #4
  70 MOVE "m";1;"data" TO #3
```

and line 70 would even print out the contents of the new file on the ZX printer for you!

You can use streams to make many other useful connections, too. If you're interested, glance ahead to Chapters 8 and 9. For the moment, it's enough to understand the general principles. Believe it or not, those same principles are used by many of the disk drive systems now available for the Spectrum – and some even use modified versions of Microdrive commands.

So who needs disk drives?

Microdrive is an excellent system for the ordinary user. It's cheap, flexible, and fast, and it gets over most of the major problems of cassette storage as well as giving you a whole host of new features. However –

- It doesn't identify file *types* with the CAT command – only file names. If you've forgotten whether 'Fred' is a BASIC program, a string array, a number array or a machine code routine, then hard luck.
- It is possible to wipe a cartridge accidentally.
- Many of the most useful commands will not work with files loaded using conventional LOAD operations.
- At £4.95 a time the cartridges are very expensive.

On several occasions I have been trying to use Microdrive and finished up with a locked-up computer and a continuously purring drive unit. Obviously one can't remove the cartridge in this situation, so the only solution is to switch off the power – and risk losing everything on the cartridge. Sinclair themselves are aware of these problems, and there are rumours that the Interface 1 ROM is being revised, but I have found the system so useful as it stands that I don't regret buying it. However, it does make me look very carefully at any more expensive storage system. Given that you're willing to spend the £300-odd that most disk systems are going to cost you, what are you getting for your money? The answers should be:

- Speed (and you thought Microdrive was fast!).
- Greater storage capacity.
- Greater flexibility in use.
- Proven reliability.

Disk drives – what they do and how they do it

The difference between a disk drive and Microdrive, broadly speaking, is the difference between a cassette tape and a gramophone record. If you want to find a particular piece of music on a cassette, you will have to wind the tape through until you get to it. If you want to find it on a record, you can usually do it by picking up the stylus and lowering it again in the right place.

A disk storage system uses a *read-write* head instead of a stylus and a flexible mylar disk coated with magnetic oxide instead of your favourite album. Unlike your favourite album, the disk is kept in a protective sleeve all the time, and this sleeve cleans it as it rotates. The system can find a given file rather faster and more accurately than you can find a track on the album because part of any disk is reserved for the *directory*, a detailed list of what's on the disk and where it is. To map out the disk the system divides it into *sectors* (rather like the slices of a cake) and *tracks*, a series of concentric rings on the disk. The sectors are marked by holes punched in the disk. Everything you put onto the disk is assigned one or more sectors of a given track. So, when you want anything, the system simply consults the directory and lines up on the right part of the disk almost immediately.

As long as the system is reasonably efficient, this means that you can get to files on the disk either *randomly*, as you need them, or *sequentially*, in a fixed order determined either by the files themselves or by programming. True random access is something you don't get even with Microdrive. Ideally, you can achieve this control with a few short, simple commands, perhaps similar to the BASIC ones you already know. With a really good disk drive system you almost seem to be operating a 200K RAM computer! However, disk drives are subject to the same bugs and jitters as any mechanical device anywhere, so don't get carried away. You need to look at your prospective new system with a cold and critical eye.

Disk drives – what you see and what you get

There are currently four Spectrum disk systems, either available or at working prototype stage, and a fifth in development at what is expected to be a highly competitive price. You would be very well advised to check the availability and price of any system that interests you before placing an order. You'd be even better advised to try it out for yourself. To get you started, here are a few pertinent questions.

- Where does the operating system reside?
- How much on-board memory does it use?
- What sort of drives does it use? Could I use a different kind if I wanted to?
- Does the price include the disk drive, interface, cabling, and power supply?

- Does it include any disk software?
- What other software can I use?
- How efficient is it? Can it handle double-density and double-sided disks?

The operating system

The idea of an operating system (sometimes called DOS, for Disk Operating System), may be new to you. In fact your computer wouldn't work without an operating system, but because you can't do anything to it you tend not to notice it. The operating system is the built-in program that directs and coordinates all the various activities of the CPU and peripherals. (It appears as a bus station in the Spectrum Adventure in Appendix 1!) Microdrive has its own operating system built into Interface 1; disk drives need one, too, and theirs can't be built into the computer itself. Obviously a complex drive set-up could eat up a lot of your 48K this way. Find out how much working space you have when the system's up and running – and make comparisons with other systems.

Software

Nowadays a good system should include some software in the basic package. A typical selection would include a word processing program, a filing program of some sort, and a 'spreadsheet' program for accounting (useful if you're going into business for yourself but not so useful otherwise). You should be able to RUN ordinary BASIC programs without any trouble, provided that they don't use more RAM than the DOS has left you, but check with the supplier if you want to buy large machine code programs, especially programs like assemblers that use high memory (i.e. addresses above 60000).

Choosing your drive unit

Listed below are the three main standards for disk drives at the moment.

- 5¼ inch (just about the industry standard).
- 3½ inch (Sony's attempt to improve on this).
- 3 inch (Hitachi's alternative – cheaper at the moment).

(Others are available, but I wouldn't recommend them.) The size refers, of course, to the diameter of the disk. Equally obviously, a smaller disk needs a smaller drive unit, which means less wasted space for you. Many systems offer 5¼ inch as standard. Others have gone for 3 inch. Both systems offer good value for money. Typically, a 40-track single-sided single-density 5¼ inch floppy disk can accommodate up to about 200K – twice the capacity of a Microdrive cartridge – and will cost you about £1.70. This is about the smallest capacity available. 400K, or 800K double-density double-sided disks cost only about £3. However, it's not quite that simple.

Getting the most out of your disks

Efficient use of the disk really depends on how well the system as a whole – disk drive, operating system, interface and computer – actually performs. A good system can pack a disk with information, but a poor one may never let you reach its full potential. Most Spectrum disk systems will handle 40-track single-density disks. Drive units for these in 5¼ inch format come at about £150, which doesn't look too bad. However, you can expect to pay very nearly the same again for the interface and operating system.

ITL Kathmill

The prototype ITL system is a version of the Byte Drive 500 disk system recently released for the Oric. It is designed for the new 3 inch Hitachi disk drive, and it's unique in offering both a 'Spectrum DOS' and a version of the CP/M standard especially favoured by business software manufacturers. (If that doesn't sound too exciting, I'll add that many games, including the original Scott Adams adventure games, are also available on CP/M). However, the CP/M standard, like the RS232 'standard', is less of a standard than it should be, so not all CP/M software will run on the ITL, and little of it has yet been released on 3 inch disks. If you can't wait, the system will also be available to order on 5¼ inch and even on the Sony 3½ inch disks. ITL will supply the interface cased with a replacement keyboard. All cabling is supplied, and there's a buffered (i.e. protected) user port for adding other peripherals and a Centronics interface for your printer (see Chapter 8). I am told that a software suite will be provided, but what you'll certainly get is the system disk, with the Spectrum DOS on one side and the CP/M on the other. The price for the package is about £120, plus the cost of the drive unit, and ITL say it will get 200K out of a typical disk.

In CP/M mode the system is designed for use with a green screen monitor (see Chapter 10), and will provide an 80-column display.

Connection is via a phono socket at the back. Both operating systems act as 'shadow ROMs', effectively replacing the computer's own ROM, so you will not notice any loss of available memory when the system is up and running. Both are also self-booting, which means they start up by themselves when you switch on. ITL expect to give their system full Random and Sequential Access Filing on its release.

Morex Peripherals

Morex are offering another prototype system, this time based on slimline $5\frac{1}{4}$ inch disk drives. It can include up to four drive units, single or double-sided. The advertised package includes the disk interface (compatible with the RS232/Centronics interface reviewed in Chapter 8), a single-sided drive unit (boxed, with power unit), cabling, an operating system on floppy disk, manuals, and disk versions of Tasword Two (for word processing), Masterfile (for database work) and Omnicalc (a spreadsheet program). According to Morex, this should give you about 200K for just £286. Options go up to £1710 for a 3-megabyte system! The 34-way Shugart standard it uses is also compatible with the Hitachi 3 inch drives, and in theory at least the system could support these drives. The target market is the small business user.

The DOS is on the system disk in Drive 1. It's self-booting. To get around the problems that arise with machine code routines in high memory, the DOS can be replaced with a cut-down 4K 'mini-DOS'. The system commands are very friendly; most are variations on the normal Spectrum BASIC commands, but all must be preceded by PRINT #4: (so don't use stream 4 for anything else!). CAT produces a complete disk directory, which tells you exactly what sort of files you have and their size to the nearest kilobyte. The minimum size is 1K (two disk sectors). Random Access and Sequential Access Filing are promised in the near future. The maximum loading speed at the moment is 32K in 6 seconds.

New system commands include a *wild card* facility – more jargon! It means you can use the symbol ^ to replace any letter or number. For instance, suppose you've been working on a new BASIC program, and there are half a dozen draft versions on the disk called "program1" to "program 6". If you type in PRINT #4: ERASE "program^" then the disk space all these files occupy will be reallocated. (They're not actually wiped out the way they would be on Microdrive.) There are also some significant additions to the power of MERGE and other BASIC commands. It is possible to MERGE even auto-run programs (not possible on Microdrive), and to 'chain' BASIC programs so that you can run through a series of routines far more than 48K long 'one chunk at a time'. Clearing out and starting again is simple – you just press the RESET button!

Technology Research

The Technology Research system has clearly gone through a lot of development in a fairly short time. Its DOS resides in a 4K EPROM inside the interface (if you're not sure what an EPROM is, look ahead to Chapter 11), and TR say that only 128 bytes of the Spectrum's on-board memory, located below the BASIC program, are used by the system. The version I saw used 40-track single-sided 5¼ inch disk drives, and could apparently support up to three drive units (40- or 80-track, single- or double-sided). Originally the TR system used its own commands, but the latest version accepts standard Spectrum keywords in BASIC programs or as direct commands. However, these commands are rather awkward – each has to be preceded by a USR command and a REM statement. To CAT, for instance, you'd have to enter something like PRINT USR (number):REM:CAT which could get very irritating after a while! TR have tried to make their system as friendly as possible, but it's a bit clumsy, and they don't sell a packaged system – just the interface (the last quoted price I saw was £85). According to TR, the system should get 100K out of a typical disk (presumably 40-track single-density, single-sided).

Primordial Peripherals

If anyone out there, especially anyone running a business, bought the **Viscount Disk System** and **FIZ** interface from a Spectrum retail store and is wondering what to do next, I have some good news for you. Your interface was originally made by the now defunct Interactive Instruments of Leicester, and stocks of it have now been acquired by Primordial Peripherals. Primordial have produced an addition to the original operating system called LODOS, which takes the original DOS out of high memory and loads it into low memory instead. The advantage is that programs such as Tasword, which need the high memory space occupied by the original DOS, can be LOADed and used once the DOS has been transferred to low memory. If you already have the older FIZ interface you can buy a package from Primordial for just £19.95 which will convert it to the new system.

The system commands are in the form of letter-group codes. (Code 'zap', for instance, promptly does just that to a specified file). The files are called by putting their names into f$, so to erase a file called "addon" you would enter:

```
LET f$="addon"
PRINT USR zap
```

This isn't as convenient as the Microdrive system commands used by Morex and ITL, but it's an improvement on the multi-part commands needed for the TR system. FIZ DOS will only handle single-density single-sided 40-track disks, which in this system, I am told, limits the available memory per disk to just over 100K.

Advantages? It's simple to use. It exists – *now*. And it's extremely well documented, with one of the friendliest operating manuals I have read in a long time. Primordial plan to improve and upgrade the interface so I look forward to hearing good things in due course. Meanwhile their starter package, for just £79.95, includes the interface itself, which connects directly to the user port, and the system disk. Everything else, including a printer interface if you want one, is up to you, though Primordial can supply most of it at reasonable prices.

Making the choice

As usual, the choice is up to you – so keep an eye on the computer press to see how the new systems shape up to their specifications. Ask questions, make sure you see what you're getting, and make sure it's what you want and does what you want. If you plan on investing in something that costs more than twice as much as your Spectrum, and will probably outlast it, then it'll pay you to be picky.

And now, a word from your computer ...

Chapter Six

Speech and Sound Synthesisers

There you are, sitting peacefully down to an evening's computing. Without looking at the screen you key in the LOAD command – and a Dalek-like voice suddenly says 'Load! Enter! Quote! Quote!' You are just wondering if micro madness has finally set in when you see the small black box sitting at the back of your Spectrum. Experimentally, you press other keys. It responds by telling you whatever it is you've just entered – anything from 'comma' to 'randomise', though for some reason it balks at the tilde (~). This is the **Currah Microspeech**, one of the best and most easily programmable speech synthesisers for the ZX Spectrum. The key readout is optional – if you get fed up with it just tell your new friend to

```
LET keys=0
```

and you will just hear judiciously amplified BEEPs – another useful feature. You can restore it with the command

```
LET keys=1
```

These commands can also be used in programs, of course.

So talk to me!

If you really want your computer to talk to you, there is quite a wide choice of speech units on the market. However, the shortlist boils down to about four or five, and of those three actually use the same chip, which means they can be programmed in more or less the same way. Points you need to look for are:

- Sound: can you understand the words?
- Versatility: how much can it actually do?

- Programming: how hard is it to put words together, and how much help does the software give you?
- Connections: can you use the unit at the same time as other add-ons?

Price isn't really a factor, except with the DCP S-pack and the William Stuart, as most speech units cost about £30. To get a clearer picture of how the available units compare, it'll help to take a brief look at the way they work.

Saying hallo to the allophones

The Currah, like many other speech synthesisers, uses *allophones*. Allophones are electronic copies of the sounds we use in everyday speech – and you can 'simulate' English with just 64 different sounds. If that seems incredible, remember that you can write every word in the English language using just 26 letters: it's not the number of sounds that's important, it's the number of possible combinations. Once you get the hang of it, programming allophone speech units isn't as difficult as it might appear, though it helps if the software is good. But getting really good sound is quite difficult. Obviously it would be a lot easier if the actual selection of sounds was done for you, using built-in firmware that manipulated the chip on your behalf. This is the principle behind the DCP S-pack, and the result is the best speech synthesis you are ever likely to hear. But the choice of words available from a unit like this one is comparatively small – you can't assemble new combinations yourself – and you have to rely on the manufacturer to give you the words you want to use.

Choosing your unit

To give you an idea of what the available units are and do, here's a brief description of some of the most commonly advertised speech synthesisers. As in previous chapters, it should be read in conjunction with Table 6.1 to get the full picture.

The **Cheetah Sweet Talker** is a neat little box that plugs into the user port and gives very acceptable sound at a sensible volume. However, its cassette software gives very little real help to the would-be programmer, despite some nice touches of genuine humour. Like all the other units apart from the DCP and the Currah you program

Table 6.1. Speech units.

Model	Type	Sound	Through bus?	Software	Price
Cheetah	Allophone	Good	Spectrum	Average	£30.00
Currah	Allophone	Fair	No	Very good	£30.00
DCP	Programmed (limited vocabulary)	Excellent	ZX81	Excellent	£29.95 + £12.95 per upgrade chip
Fuller	Allophone	Good	Spectrum	Poor	£30.00
W. Stuart	Allophone	Poor (but can go to hi-fi)	ZX81 + audio sockets	Poor	£39.00

```
  10 PRINT "SPEECH SYNTHESISER E
XAMPLE"
  20 INPUT "Key in the OUT numbe
r used by   your unit (see text
) ";out
  30 INPUT "Key in the length of
 the pause  between allophones
(1 to 10) ";pause
  40  IF pause<1 OR pause>10 THE
N  GO TO 30
  50 FOR j=1 TO 18: READ s: OUT
out,s: PAUSE pause: NEXT j
  60 DATA 61,39,19,13,12,44,55,0
,0,33,60,0,0,14,19,33,51,0
  70 INPUT "Was that OK? (Y/N) "
;a$: IF a$<>"Y" AND a$<>"y" THE
N  CLS : RESTORE : GO TO 10
  80 PRINT ''"OUT port is ";out'
'"Pause is ";pause
```

Fig. 6.1. Demonstration program for speech units using the same allophone system as the Cheetah.

it with OUT commands such as OUT 31, 26, which call 'sound number 26' from the on-board firmware. If you'd like to do something a little simpler than delving around the LISTing of

Currah's cassette, try entering the program in Fig. 6.1. Incidentally, those of you with Interface 1 should enter the OUT value as 31 – those not yet blessed in this way can enter OUT 7. You can see similar routines at work in the Spectrum Adventure – the number codes are put into arrays at line 7300, and read back in the course of the program by the subroutine at 8000. The software Cheetah sent me was obviously pre-Interface 1, because it crashed mightily as soon as I tried to RUN it! If you're feeling heroic you could LIST its program and change all the OUT 7s to OUT 31s, but I find it a lot simpler to remove Interface 1 for a while!

Cheetah Sweet Talker

Sound:	9 (good volume, understandable as long as your programming is fairly sound)
Versatility:	8
Programming:	5 (probably the worst feature of this unit)
Compatibility:	9 (connects with most other peripherals)

Currah have been very successful with the **Microspeech**. It's simple to program, effective, and many software houses include an option for it in their games. It works by modulating signals going to your TV set; a useful idea, but the signal does interfere with TV picture and sound. Currah have tackled the problem by fitting a tuning screw on the top right-hand corner of the unit. Careful adjustment should cut the interference to a minimum, but avoid large areas of bright colour on screen.

Once you get down to actual programming you'll find you have a user-friendly system with a helpful manual and a reasonably good software demonstration. Try, for instance, entering

LET S$="ha(ll)(oo)"

and your Currah will give you a cheery robotic greeting. However, if you plan to use it with existing programs make sure that S$ isn't already in use somewhere else. If you now enter PRINT S$ then

*a(ll)(oo)

will appear on the screen. The asterisk shows that the Currah was able to handle your programming. If there are problems, the first letter will be replaced by a question mark. The Currah manual gives you a start with your speech programming, backed up with a short (too short?) demonstration routine on the cassette, and a slow but enjoyable talking adventure game.

Currah Microspeech

Sound:	7	(distorted by interference)
Programming:	9	(probably the easiest and best documented)
Versatility:	10	(on-board firmware and simple programming make this unit ideal for many applications)
Compatibility:	4	(can only be the last add-on in the chain)

The **DCP S-pack** is a Rolls-Royce speech synthesiser. The sound it produces is far better than any other unit I have heard – but at the price, it should be. The DCP uses a pre-programmed vocabulary, as outlined above, so the words are as good as they can be; to achieve the same quality on, say, the Fuller, would take a great deal of time and hard work if it could be done at all. But there's a snag. The basic unit (which costs a competitive-looking £29.95) has just one out of a possible four chips, and that one chip contains just 72 words. Of those 72 words, 31 are numbers, 26 are individual letters of the alphabet, five are programmed silences, and two are simple tones. The remaining one says, with remarkable clarity, 'This is Digitalker'. Frankly, I don't find this a particularly useful selection, though it does at least allow the computer to say 'OK' when it has accepted a command.

Three extra chips are available, each extending the vocabulary by another 72 words, but they weigh in at £12.95 each, and the choice of words is rather arbitrary – how could it be anything else? Furthermore, chips three and four will only work together. For quality, the DCP is unbeatable, but it will only be a good deal for those who find its vocabulary useful. To check, write to DCP and ask for a list so that you can make up your own mind.

DCP S-pack

Sound:	10	(indisputably the best)
Programming:	10	(could hardly be easier)
Versatility:	4	(extremely limiting)
Compatibility:	8	(only with other peripherals that have ZX81-type connectors)

The **Fuller Orator** is available as a separate unit, or as part of the impressive Master Unit which also includes a three-channel sound synthesiser, a joystick interface and a cassette interface. Using the Fuller's through bus you can connect almost any other peripheral: so ten out of ten for compatibility. However, the documentation is poor – little or no attempt has been made to make the Orator user-friendly. The software is a selection of remarks put together to demonstrate the unit at a ZX Microfair: again, I only found out how to get useful sounds by breaking into this program and LISTing it. To save you the trouble, Fig. 6.1 can again be used for an example of this unit at work: enter 159 when you're asked for the OUT value, and experiment with different pause values.

Fuller Orator

Sound:	8	(powerful and reasonably clear)
Programming:	4	(again, the worst feature)
Versatility:	8	(lacks the Currah's enormous range of possible applications, but could still be extremely useful)
Compatibility:	10	(most other peripherals can be fitted)

The program listed above will also work with the **William Stuart Chatterbox**, a rather more sophisticated and decidedly more pricey unit. One of the interesting-looking DIN connections at the top acts as an interface for the Big Ears speech recognition unit (of which more anon); the other allows you to take output from the unit to your hi-fi system. A small jack socket between them allows you to connect the unit with the MIC socket on your Spectrum so it can act as a BEEP amplifier. When this unit is connected with the William Stuart Music Synthesiser, which is sold as a bare-board add-on, it will also amplify the output from that, so what you are paying for is potential versatility rather than actual performance.

The sound of this unit is not good – it's loud, and you can't control it. The manual, too, is rather difficult to get along with, and the printed sample program is a wee bit complex. William Stuart's own software includes a talking version of 'Hangman' which, like many games of this sort, tends to run rather slowly, and a rather poor game called 'Chromacode'. However, the Chatterbox is worth considering if you're interested in other William Stuart products.

William Stuart Chatterbox

Sound:	5	(low quality – but at least you can feed it through to your own hi-fi)
Programming:	3	(little good software support, poor manual)
Versatility:	10	(many other possible applications)
Compatibility:	9	(many other add-ons use the narrow connector)

Using speech synthesisers

The Cheetah, the Fuller, and the Chatterbox all use the same chip, so the allophones available are exactly the same and use exactly the same codes. Any of these units will work with the Spectrum Adventure in Appendix 1. Remember to alter the OUT command in line 8020 to OUT 159 for the Fuller Orator – and if you're using the Cheetah without Microdrive.

The Spectrum Adventure probably seems a very frivolous use for units like these, but the Currah in particular is almost designed to be frivolous. Besides, even my use of them does suggest a more serious application. In the Spectrum Adventure, the computer will only 'speak' a valid command. A similar piece of coding in a serious business program could be a handy check, ensuring that the user *knows* that a valid command has been safely entered before going on to the next stage. A unit like the Currah is particularly useful because it confirms each individual keystroke. It's too slow to keep up with a touch typist, but a non-typist who needs to look at the keys a lot might find it comforting to hear the 'right' command being entered – and useful to hear if the *wrong* key has been pressed, too.

Talking back

So far we've been holding rather one-sided conversations with the Spectrum, limited to what you can program into one or other of the speech synthesis units currently on the market. But suppose *you* could talk to your Spectrum – and it could 'understand' what you said?

Well, it's possible – up to a point. Getting a computer to recognise speech is not too easy at the best of times, and even if it does it's only going to recognise the voice of the person who programmed it. (Such touching fidelity ...!) Units like this work by analysing the sounds you make and converting them into numbers. The gadget that does it is called an *analogue to digital converter* which basically means anything that turns non-number information such as voltage variations into numbers. Typically a unit will ask you to say the same word several times, average out its readings each time, and come up with a formula that will allow it to recognise the same set of sounds next time around. Not surprisingly, you can usually fool it with similar-sounding words (like 'wine' and 'dine'), and if your voice happens to be a bit off after a night of wining and dining your faithful computer may totally ignore you. (Who said computers aren't like people?) To program any speech recognition unit you need to work in a quiet room, and speak very clearly and consistently. That takes a little practice, so be patient. There are two systems currently available.

Big Ears, from William Stuart, is probably the better unit – but at nearly £58 it's extremely expensive, and it will only connect to the Spectrum if you use either the Chatterbox unit reviewed above or a special input/output connection on the music synthesiser reviewed later in the chapter. Measured against the competition, all three of these units are overpriced. However, in a quiet room I was able to get remarkably accurate results from the test program.

Big Ears arrives complete with microphone, microphone stand, and the necessary connections to the other William Stuart units. The manual, however, is both inaccurate and obscure. When you've sorted it out you can alter the number of times the program asks you to repeat a word (the more the better), and the number of words it can have in its vocabulary. However, you are not really told how to get this vocabulary working in your own programs. This is not easy, and you will need both time and patience to get the results you want. However, the unit can cope with quite a large number of words (certainly enough to operate the Spectrum Adventure!) so the possibilities for serious applications are good.

Micro Command is both cheaper and better documented, but with a maximum vocabulary of 15 words it's a bit limiting. It comes as a typical Spectrum add-on black box that plugs directly into the user port, but that's it – there's no through bus, so if you're using other peripherals this must be the last one in the chain. Like Big Ears, it arrives complete with a microphone that plugs into a jack socket on the top of the main unit.

Micro Command, too, is far from infallible – it's up to you to speak your commands consistently, and to keep background noise to a minimum. The manual itself is less useful than the 'additional information' also supplied, but you should find it reasonably easy to understand, and the software is rather more entertaining than William Stuart's. If you're buying a unit for fun, this is the one for you. If you're interested in more serious applications, then it's probably worth paying the extra for Big Ears, perhaps as part of the full interactive speech system from William Stuart.

Using speech recognition systems

The main problem with systems of this kind is the time any unit takes to recognise a word and respond to it. The answer is to structure your program so that there are only a few words to choose from at a time, preferably as different from each other as possible. If you're using the unit to choose options from a menu (like the options in the Spectrum Adventure or, more seriously, the options in a more complex accounting program), you'll find that this sort of structure makes sense in any case. You should also allow for Murphy's Law and include an option that allows the user to start again, or return to the previous menu – misunderstandings are fairly frequent!

But perhaps the most exciting application for this type of unit is to team it up with some of the control devices discussed in Chapter 11. I used it with the REL 4.2 unit from Harley systems and spent an entertaining hour or so switching lights on and off with pure voice power. This is the perfect microcomputer application for the incurably lazy – like me – and for people with handicaps who might have rather more of a problem operating a conventional switch. Using the William Stuart system the speech unit could confirm that your command has been understood!

Three-part harmony

It is entirely possible that the idea of a talking computer leaves you cold. Musical computers, on the other hand, are almost commonplace, and it will come as no surprise to learn that there is a wide choice of sound and music synthesisers for the Spectrum. As with the speech units, many of them are based around the same chip, and can be programmed in almost exactly the same way. The changes rung by different manufacturers range from a bare-circuitboard model at £19.95 to the top-flight version of the Fuller Box, the Master Unit, which weighs in at £54.95 and includes the Fuller Orator, a joystick interface, and an electronic self-switching cassette interface. As with the speech synthesisers, what you're really looking for is

- Good sound quality
- Easy programming
- Compatibility with other units
- Value for money

Value for money really becomes an issue with these units – the price range is wide, and depending on your own abilities the best buy won't always be the expensive unit in the plush box. Software's important, too – it isn't fair to expect the new user to understand and handle the 14 different variables that normally have to be programmed, so my ratings once again include a separate score for software, which ranges from the above average to the mediocre. As usual, you should check out products that interest you in Table 6.2 to get the full picture.

Table 6.2. Sound synthesisers.

Model	Sound	Programs	Through bus?	Joystick port?	Price
Add-On	Good	Good	No	Yes – pot. and Atari	£19.95
Bi-Pak	Good	V. poor	ZX81	No	£32.75
Fuller Box	V. good	Poor	Spectrum	Yes	£29.95
Petron	V. good	V. good	ZX81	No	£24.95
Signpoint	V. good	V. good	ZX81	No	£28.00
W. Stuart	None (goes to hi-fi)	Fair	ZX81	I/O port	£30.00
Timedata	V. good	Fair	ZX81	Yes	£29.95

Signing the registers

Before any detailed reviews, it'll help you to know something about the chip that's used and the way it's programmed. To help you on your way we've included a program (Fig. 6.2) that should make it

```
  10 PAPER 1: INK 7: BORDER 1: C
LS : PRINT "THE SOUND PROGRAM"
  20 INPUT "Enter the OUT number
 your unit  uses to select a re
gister."'outreg
  30 INPUT "Enter the OUT number
 your unit  uses to send inform
ation to the registers."'outval
  40 CLS : LET temp=0: LET place
=0: DIM r(14): DIM s(14): DIM m
(14)
  45 FOR j=1 TO 14: READ m(j): N
EXT j
  47 DATA 255,15,255,15,255,15,3
1,255,16,16,16,255,255,15
  50 RESTORE 60: FOR j=0 TO 13:
READ r$: PRINT AT j,0;j: PRINT
AT j,3;r(j+1): PRINT AT j,9;r$:
NEXT j
  60 DATA "A channel fine tune"
  70 DATA "A channel coarse tune
"

  80 DATA "B channel fine tune"
  90 DATA "B channel coarse tune
"

 100 DATA "C channel fine tune"
 110 DATA "C channel coarse tune
"

 120 DATA "Noise control"
 130 DATA "Noise/tone selector"
 140 DATA "A volume"
 150 DATA "B volume"
 160 DATA "C volume"
 170 DATA "Waveform fine tune"
 180 DATA "Waveform coarse tune"
 190 DATA "Select waveform"
 200 PRINT AT place,3; FLASH 1;t
emp;"  "
 210 IF INKEY$="6" AND place<13
THEN  LET place=place+1: PRINT
AT place-1,3;r(place);"    ": LE
T temp=0
 220 IF INKEY$="7" AND place>0 T
HEN  LET place=place-1: PRINT A
T place+1,3;r(place+2);"    ": L
```

```
ET temp=0
 230 IF INKEY$="5" AND temp>0 TH
EN  LET temp=temp-1
 240 IF INKEY$="8" AND temp<m(pl
ace+1) THEN  LET temp=temp+1
 250 IF INKEY$="0" THEN  OUT out
reg,place: OUT outval,temp: LET
 r(place+1)=temp
 260 IF r(8)=255 THEN  DIM r(14)
: CLS : GO TO 50
 270 GO TO 200
```

Fig. 6.2. Sound unit demonstration program.

easier to use any of these units. Have a look at it – most of it's actually taken up with the text needed to make it easy to use! Enter it properly and you'll get a screen display like the one in Fig. 6.3. The first figure in each line is the *register number*; the next figure shows the current value of what's in there (all yours will start at 0), and the text tells you what it's for.

```
0   0      A channel fine tune
1   0      A channel coarse tune
2   0      B channel fine tune
3   0      B channel coarse tune
4   0      C channel fine tune
5   0      C channel coarse tune
6   0      Noise control
7   0      Noise/tone selector
8   0      A volume
9   0      B volume
10  0      C volume
11  0      Waveform fine tune
12  0      Waveform coarse tune
13  0      Select waveform
```

Fig. 6.3. Screen display from sound unit demonstration.

There are fourteen *registers*, and as far as you're concerned they simply hold numbers that you can enter using the cursor keys. Move up and down to find the one you want, move left to reduce the number or right to increase it, and press 0 to enter the new number. If you have a joystick, try adapting the program to work with it.

What I've called 'channels' A, B, and C control the *pitch* of the sound – i.e. whether it's high and thin or low and growly. You can change the pitch by altering registers 0 to 5, and the volume by altering registers 8–10 – so when it's 0 you won't hear anything! However, if you set one of these registers to 16 then it comes under 'waveform' control. This will alter the 'shape' of the sound (producing, for instance, warbles and trills) and you can program it using registers 11–13. Odd numbers in register 13 will give a short sound followed either by silence or a sustained tone. Even numbers give a regular pattern of sound. Registers 11 and 12 control the variation of the sound – if you like, the size of the wave!

Yes, I've left two out. Register 6 lets you play about with the noise generator. To get the most out of register 6 you also need to set register 7 fairly carefully, so make a note of the results you get with different settings. The only way to appreciate what these registers do is to switch the channels to noise (rather than tone, which makes *nice* noises) and try entering different values into register 6. Have fun!

Making your choice

The **Add-On** looks unpromising – it's a bare circuit board with a small speaker unit attached. But it really does deliver the goods! As well as the sound generator it includes a joystick interface with a D-type socket and two sockets for potentiometer joysticks. It arrives complete with a simple but quite effective demonstration and sound editing program from Program Power. Sound from the built-in speaker is remarkably powerful. The volume control looks a bit odd, but gives an acceptable range of sound. The connections for the potentiometer joysticks are somewhat fragile, but the D-socket sits solidly on the board. An extra lead can be plugged into the Spectrum EAR socket (allowing the unit to work as an extremely effective BEEP amplifier) or into the socket at the top of the Add-On itself, where it amplifies signals from the on-board chip.

The duplicated manual supplied is a little unfriendly at times, but the information it gives is reasonably clear, particularly when you use it in conjunction with the software. An added bonus on the Program Power tape is a potentiometer joystick demonstration (rather basic, but effective), and a machine code routine for reading two potentiometer joysticks and their fire buttons. This is the kind of service one is always hoping for in the computer business! The only real snag with this unit is the lack of a through bus.

Add-On

Sound:	7	
Programming:	8	
Compatibility:	8	(despite the absence of a user port, the joystick interface makes up for a lot!)
Value for money:	9	

The **Bi-Pak ZonX** arrives in a neat black case with an accessible but slightly stiff volume control. It was designed for the ZX81, so make sure that you connect the extra **PCB** between this unit and the user port of your Spectrum, or it will steadfastly refuse to respond to anything you send to it! The price is a little high, so it's rather a shame that the manual is still living in the Dark Ages before Spectrum. If you want any real idea how to use the ZonX, start off by turning to page 22, which introduces the novel idea of OUT commands. (OK, I know the ZX81 didn't have them but surely with a potential market of about 2 million it's worth writing a separate Spectrum manual?) There's no supporting software, alas, so the manual is all you've got – hence my annoyance. It's a pity that a good, functional unit should be let down in this way. Without better software support this is not really good value for money.

Bi-Pak ZonX

Sound:	7
Programming:	4
Compatibility:	7
Value for money:	5

The **Fuller Box** also has a music synthesiser as one of its many options. Again, the high quality of the hardware and the excellent design are let down by poor documentation and software, but make no mistake – this unit is good, and its facilities include a full-width through bus, a self-switching cassette interface, and a joystick port, as well as strong sound from the built-in speaker. Only the volume control is a bit primitive – you have to fiddle around at one side with a small screwdriver – and I found it a bit worrying to open up the case, which you must do to feed through the aerial lead from your TV to the back of the Spectrum. With the speech synthesiser included, the price of the Box comes up to £54.95, but William

Stuart charge that for Chatterbox alone. It's a good buy – but you'll need to read this chapter carefully and do a lot of experimenting if you want to make full use of it.

Fuller Box

Sound:	9
Programming:	4
Compatibility:	10
Value for money:	10

The **Petron Trichord** is an attractive boxed unit with a good, solid volume control and very pleasant sound. The demo tape includes a very competent performance of a Bach cantata! At £24.95 the Trichord leaves several other units standing, and software support is admirable – Petron have made a real effort to give the user simple, flexible music and sound effects programming. The tape starts with a simple machine code routine that gives access to 255 different pre-programmed sound effects. Nothing could be simpler, and the effect can be called from a program at any time after the machine code has been LOADED. If this doesn't satisfy the avid sound effects creator, there's also a useful toolkit routine later on the tape that allows you to make your own noises.

However, the unit really comes into its own for music programming. The Trichord owner can program his composition directly using simple software and a genuinely useful manual. The only better software I have seen comes with the Signpoint unit reviewed below. It isn't perfect, perhaps because the idea is to use as little memory as possible; personally, I'd have welcomed a slightly more elaborate display. However, it's a giant step forward from the competition and is rated accordingly.

Petron Trichord

Sound:	9
Programming:	9
Compatibility:	8
Value for money:	10

The **Signpoint Sound Synthesiser** is slightly more expensive than the Petron, but offers very similar features – good sound, with a decent volume control, pleasant casing and, above all, user-friendly programming. Of the two I slightly prefer the Signpoint, because of its clear music programming display, but on all-round use it still loses one or two points to the Trichord. The manual, though very cheaply produced, contains genuinely useful information, including a table of musical notes and their frequencies together with the appropriate register settings on the unit. A good product and good value.

Signpoint Sound Synthesiser

Sound:	9
Programming:	9
Compatibility:	8
Value for money:	10

The **William Stuart Mk 3 Synthesiser** costs £30, and it takes nerve to charge that for a bare-board add-on when the same money will get you the elegantly-designed Fuller Box. However, this one also includes an input/output board and a DIN-type connection for your hi-fi or for the Chatterbox and Big Ears units. No software seems to be supplied with the synthesiser, and the manual is horrifyingly technical. The Composer program, which William Stuart market separately, is very rudimentary and rather easy to crash. More interesting is the ARP program, which converts your familiar Spectrum keys into the keyboard of a fairly powerful synthesiser, though this one is also rather easy to crash. Just persevere! The input/output board will really only be of interest to serious and knowledgeable users, so my ratings are based on the unit's other connections. Incidentally, the connection to the computer is extremely wobbly.

William Stuart Mk 3 Synthesiser

Sound:	none (uses external amplifier)
Programming:	8 (but must be bought separately)
Compatibility:	8
Value for money:	8

The **Timedata ZXM** is another attractively cased unit with a good, solid volume control. It may seem slightly pricey but it does include a joystick port, so it's still pretty competitive. The sound is pleasant, strong, and easily controllable. Software support is a bit thin, but the 'Super Editor' program has some genuinely useful features. The screen display allows you to enter up to six different sets of numbers into the registers,and this is very handy. Unfortunately, there isn't any room left to show what all the registers are for! Still, the manual gives quite a reasonable explanation, and if all else fails there's always this book! I like the ZXM, and for all-round use I'd rate it highly, but for serious music enthusiasts the Petron and the Signpoint are better buys.

Timedata ZXM

Sound:	9
Programming:	6 (could be better)
Compatibility:	9 (joystick port and through bus)
Value for money:	9

Choosing a sound unit

Your final choice will depend very much on the work you want to do. If you're interested in a complete interface and sound system, then the Fuller is a worthwhile choice provided you can handle the programming, and the William Stuart opens up a good many interesting options. If you're serious about music programming, both the Signpoint and the Petron will give you an easy start. If you want general game sound effects and a useful joystick port, try the ZXM. I hope these short reviews will help you to make a sound choice.

Chapter Seven
Light Pens and Digital Tracers

Are you a frustrated artist? Have you ever felt that you might have a talent for drawing and design if only someone would give you a chance? Or, perhaps more likely, have you ever felt that you could do a better job on graphics than most of the games designers? If so, then the good news is that your computer can now be fitted with some powerful graphics hardware that will make the task of any microchip Michelangelo that much easier. And it can all be done without breaking the bank.

The graphics file

For the Sinclair Spectrum there are three different types of graphics hardware currently available:

- Light pens
- Digital tracers
- Graphics pads

A *light pen*, at least in an ideal world, is a handy little gadget that allows you to 'draw on the screen' with a device very similar to a pen. As well as drawing pretty pictures you can also use them to 'point out' items on a screen menu – especially useful for business applications or dotty adventure games (see Appendix 1!). The *digital tracer* is a delightful device that looks rather like the 'pantographs' I used to buy to help me copy drawings in a larger or smaller size. And effectively it does just that, allowing you to copy flat artwork of any size (within reason) directly onto the computer. You can also create original artwork directly, or from your own rough drawings. The *graphics pad* is rather like a combination light pen and digital tracer, but the pen, instead of operating on the screen, is used on a sheet of glass or perspex mounted over a special sensing surface. As the pen is pressed down, the tip is pushed back into the barrel and operates a microswitch. The pen 'reads' a crosswire grid rather like a super-fine version of the grid underneath the Spectrum keyboard. The result is a screen image that is an extremely accurate copy of the movements you have made across the pad.

Introducing the light pen

To understand how a light pen works you need to understand how a TV picture is actually built up on screen. The 'picture' is actually no such thing – it's the trace left by a dot of light (or three dots in the case of a colour TV) tracking backwards and forwards across the screen so fast that it manages a complete scan in one twenty-fifth of a second. That's fast, but the Spectrum is faster – in that time it could carry out up to 140000 operations! The light pen acts as a sort of electronic stopwatch – it can detect where it is on the screen by checking how long it takes the scanning dot to reach it! If its response is accurate (and that will depend on how good its software is as much as anything else) the result is a pair of x and y coordinates that you can use as you wish – to fix a point on screen in the memory, to drawn an INK pixel there, or to check which display line the pen is pointing at so that you can send your program to the chosen subroutine. If all this sounds complicated, it is – but not for you. If you're programming in BASIC for a light pen, you're usually dealing with a single number that can be treated just like any other variable.

However, if a light pen is to do any of these jobs it must be

- accurate – stray light can affect the reading;
- supported with fast and well-written software;
- easy to use in your own BASIC programs.

To make sure the pen is accurate, the software will normally include a *calibration routine*. This checks that the reading going back to the computer agrees with what you can see. If it doesn't, then the neat black line you are trying to draw could appear three inches up on the left-hand side. Calibration is important!

Making the choice

The light pen marketed by **Custom Cables International**, is *not* another version of the dk'tronics pen, as some reviews have suggested. It's a completely different design, with completely different software. At £15 it's the cheapest of the three, but also the most limited. However, it performs very creditably on every option offered, and these include all the geometric shapes supplied on the others plus a nice extra – a polygon. You choose the number of sides for yourself, and you can change the number with a POKE command to draw others in the course of the program. It will store and display screens, but there is no menu option and no freehand option. However, it's good value if you don't want these.

The **Trojan** light pen is advertised (at the time of writing) with a picture of someone drawing a highly-detailed sports car directly onto a TV screen. I note that it is an artist's impression, probably because anyone who could do it would have to have a hand of iron and the patience of a saint. The main problem here is the software. The calibration routine is inaccurate, and a bug in the menu demonstration makes it read the pen position far too quickly. As a result, the program told me that my Spectrum does *not* use the Z80A chip, which I'm sure would be news at Sinclair Research! The freehand drawing routine works so slowly that it is effectively useless. It is just about possible to draw circles, lines and other pre-programmed figures, but the pen is too inaccurate to make this easy. You're better off with a joystick and some good graphics software. The cost is £17.25.

The **dk'tronics Mk. 4** light pen is a very different story – but the number is important. The Mk. 3 version is little better than the Trojan pen but the Mk. 4 does, in fact, allow reasonable freehand drawing. The only problem is keeping your hand steady on an almost vertical surface. (Maybe you should set your TV screen into the top of the table!) I like this pen a lot – enough to include it in the Spectrum Adventure, in fact. Besides creating lines, rectangles, circles and arcs, you can store up to five screens and bring them up in rotation at speeds varying from remarkably fast (pseudo-animation, yet!) to remarkably slow. The freehand drawing routine takes a little getting used to, but is well worth the effort. It's excellent value at £19.95, and the rest of this section deals with programming it.

Using the light pen

Using the light pen in your own programs is a lot easier than you might think. The manual includes detailed instructions for saving the necessary code, but I saved mine to Microdrive straight away. The light pen picks out items on a screen menu by identifying the screen row (from 0 to 21) that the pen is pointing at. If there are only one or two options this is easy enough – you can, for instance, put in lines such as

```
100 PRINT AT 8,0;"1 Look for an exit"
110 PRINT AT 10,0;"2 Enter any other command"
```

Using the short three-line subroutine suggested by dk'tronics (which I put at line 9000 in the game) you can then use the variable Lno as follows:

```
120 GOSUB 9000: REM Lightpen subroutine
130 IF Lno=8 THEN GO TO 5000: REM Exit display
140 GO TO 2000: REM Command routine
```

You don't need to specify the variable value in line 140: if Lno isn't 8 then you obviously don't want to look for an exit! All the same, if this makes the routine rather too liable to drop you into option 2 you might prefer something like

```
140 IF Lno=10 THEN GO TO 2000
150 IF Lno=0 THEN GO TO 120
```

I've included this routine because it shows how a few lines of code can speed up your program immeasurably – especially if you're not a typist. It's a technique that's particularly useful when combined with the fast filing capabilities of the Microdrive. For instance, the menu could easily be used to call up a file from Microdrive, alter something in the file, and then store the updated version – all in a couple of minutes, even for a fairly long file!

A trace of artistry

The light pen is a useful graphics tool and a very useful menu selection device, but one thing it can't do is to copy images directly from flat artwork. If you want to create detailed graphic images using the full resolution of the Spectrum, you'll need something else, and the cheapest available alternative is the **RD Digital Tracer**.

Figure 7.1. is an example of what it can do, and the way it's arranged for use is shown in Fig. 7.2. The principle behind it is comparatively simple: there are only two moving parts, and both contain a

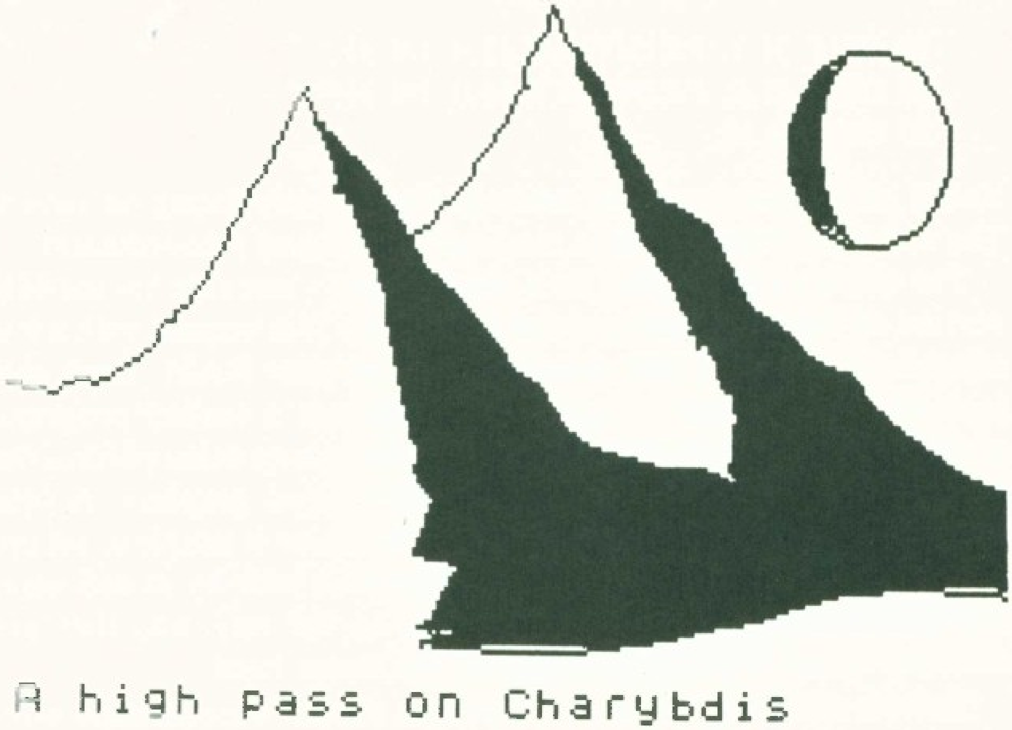

Fig. 7.1. Artwork produced in a few minutes with the RD Digital Tracer.

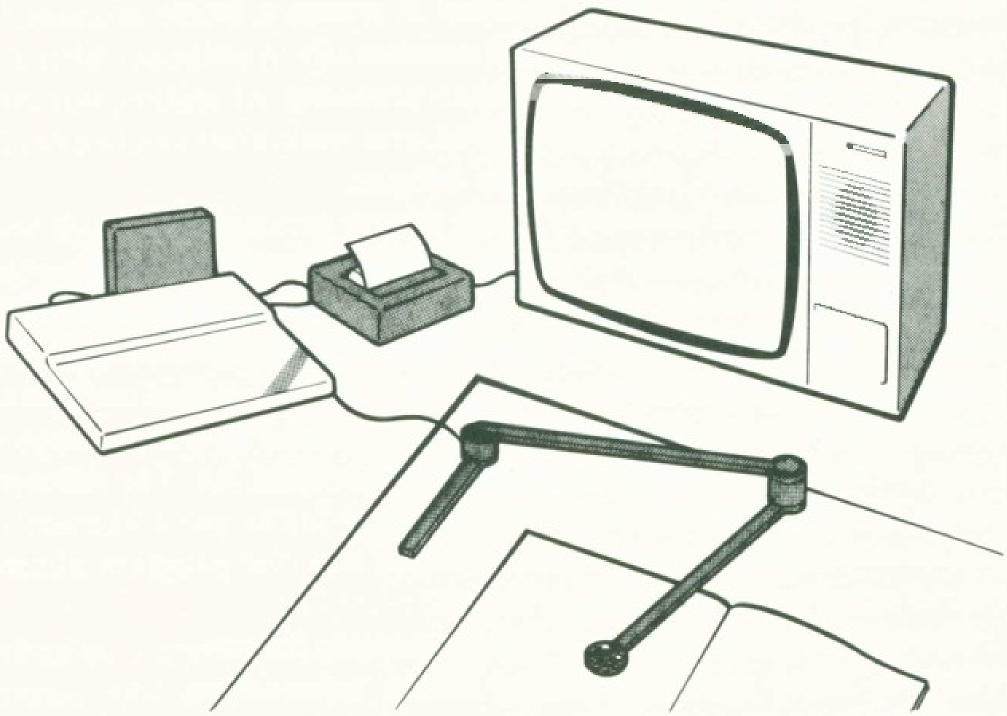

Fig. 7.2. The RD Digital Tracer in use.

potentiometer (which, you will remember, is also used in some joysticks). If you've bought the tracer and you want to see the effect of the two potentiometers for yourself, then try the following program with the tracer connected:

```
10 OUT 31,1: PRINT AT 12,0; IN 31;" "
20 OUT 31,0: PRINT AT 14,0; IN 31;" "
30 GO TO 10
```

You'll get a printout of the readings they are producing on screen. The software converts these readings into x and y coordinates.

The software is supplied on a cassette tape – one side has the 48K version of the programs and the other has a 16K version with a few alterations to make allowances for the smaller RAM. Each side includes four routines – a basic 'draw' program, a 'scaling' program, a 'retrace' program that lets you copy a shape onto other parts of the screen, and a routine for creating user-defined graphics. A fifth program combines all the other four. The tracer is simple to set up, and simple to use. My sample was let down a little by its software, but RD tell me that a new cassette is on the way, and they are working on an improved version of the tracer itself.

Besides allowing you to trace outlines, the software includes some simple routines for filling, colouring, and deleting, though the fill routine in particular was rather difficult to use on my sample. In fairness, this was probably because it makes a brave attempt to prevent 'leakage' through gaps left accidentally in outlines – after all, there's nothing more frustrating than watching ink spilling out into an area you meant to be paper colour, and wiping out any line detail you might have left there into the bargain.

Tracing outlines accurately takes a little getting used to – if you move too fast the outline on screen tends to 'cut corners' because the software can't absorb all the coordinates you're plotting quickly enough. The other problem is shaky hands! The tracer is remarkably sensitive, and even small movements tend to be magnified on screen! The answer is to use the scaling program supplied by RD and copy a fairly large image – ideally a good bit larger than A4. The reduced image that appears on screen should be a fairly good likeness if you're reasonably careful – and if you're not, the program allows you to erase small areas quite easily.

The tracer arrives complete with a template for setting it up and a reference sheet giving you very useful information like screen coordinates and memory areas for every part of the display. If you're assembling the Spectrum Adventure you might like to include some graphics in the top third of the screen; as written, the program leaves space for them. When you've drawn them you can transfer the display to Microdrive by BREAKing into the program and entering the following command (without a line number):

SAVE "title" CODE 16384, 2048: SAVE "titlea" CODE 22528,256

The first half of this line records the individual pixels that make up the outlines and ink fills on your display: the second half records the attributes, the computer's record of the colour combinations in each character square. Notice that the whole thing only takes up just over 2K, so you could easily get about thirty of these onto a Microdrive cartridge. You could even box clever by having the same basic display used with different sets of attributes (for instance, black squares replacing a white opening in a black wall). To make it easy for the program to access these files, you can replace "title" with the relevant location number, but don't forget to add the "a" to the end of the second file name; the Microdrive won't accept two files with the same name.

Once you've started to explore the tracer software you'll soon begin to see other possible applications. For instance, suppose you replaced the PLOT commands with a series of adventure game subroutines called by different values of x and y? That way you could lay a map – an ordinary Ordnance Survey map, say – under the tracer, and do a complete orienteering exercise! I can see it now. 'You are on the edge of a high cliff. The mist is coming down. What do you do?' Quite seriously, this could be a valuable educational exercise – and you could line up the tracer on the map very accurately by using a modified version of the scaling routine, which lets you plot precise values for the corners of a document into the computer's memory.

As you will have gathered, I find the RD Digital Tracer extremely hard to resist. If you're in the least bit serious about graphics it will be an invaluable addition to your armoury. The promised improvements should make it an even better deal – but a tool like this for around £55.50 (or £75.50 for the 'professional' version, which can cope with larger artwork) is a bargain by any standards.

Grafpad – the computerised sketchpad

Last, but by no means least, is the **British Micro Grafpad** (Fig. 7.3). I

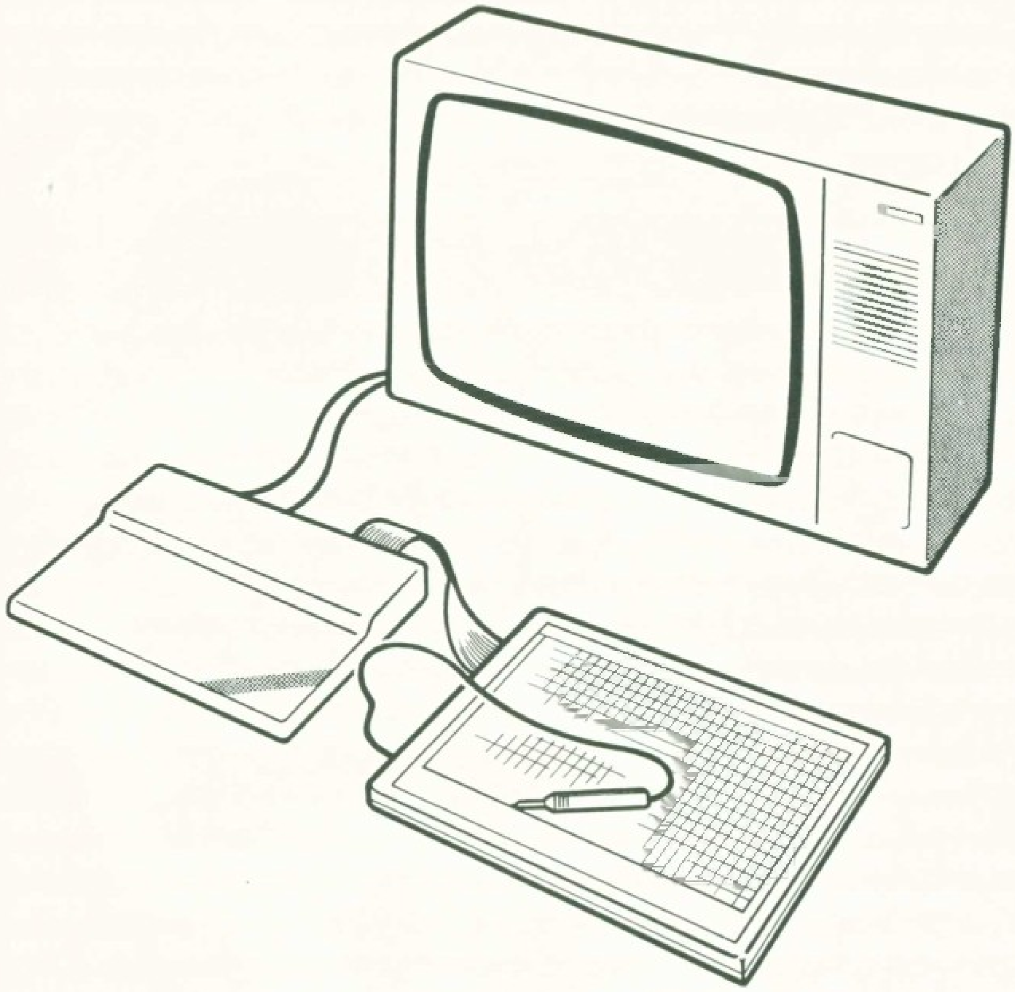

Fig. 7.3. The British Micro Grafpad.

have placed it late in the chapter because of its asking price, currently £143.75. Why, you might ask, should you pay the price of your computer for a peripheral? The answer's very simple – because buying Grafpad is like buying a completely new computer. The unit arrives complete with a full manual, all necessary software, the Grafpad, the operating pen, and all necessary connections for your Spectrum. When I tried it out at British Micro in Watford it was connected to a Cub Monitor (see Chapter 10) and the effect was staggering. The accuracy and detail that are possible with this system make using it a pure joy (see Fig. 7.4). My main problems were getting to grips with the new keyboard (the kit includes a keyboard overlay to help you work out the dozens of new functions made possible by the software) and some initial uncertainty with the pen itself.

The pen contains a microswitch so that it will only operate when you press gently down on it. What happens next depends on which

Fig. 7.4. High-resolution artwork produced on the Grafpad.

of several possible modes you are in. You can use the pen to define points for geometrical shapes – in this case moving it around the Grafpad will produce a shape that varies according to the pen position. When you get one you like, a keypress will fix it on the screen display and in memory. Or you can use it freehand – a fast, flowing stroke is best in this case, as the pen is slightly sensitive to jittery hands. Switch to another mode and the pen becomes a paintbrush, accurately filling in selected areas with your chosen ink colour. In yet another mode it's an eraser – and because it's a pen it's easy to be accurate with it.

British Micro software support for the Grafpad is excellent. Already available is a computer-aided design program that allows you to create and store a battery of user-defined shapes, take them from memory, and place them anywhere on the screen. The program costs about £20 and, if you want it, it's worth it. The possible applications are almost limitless – anything from designing electronic circuitry to drawing up detailed architectural sketches. I also saw a fascinating perspective program under development – just the thing for people like me, who never seem to get perspective right without outside assistance.

To go into greater detail about the Grafpad would only confuse the issue – it takes time and practice to familiarise yourself with its many facilities, and the best way to see what it can do is to watch it being demonstrated by someone who has already mastered it. Fortunately it has been making many exhibition appearances recently, and I have no doubt it will make many more. Go and see it for yourself.

The Grafpad is not a toy. It's a very clever and carefully-designed concoction of hardware and software that will be a boon to professional designers and technical artists, and pure joy to the talented amateur. If your talents don't really lie in this direction, forget it – you could finish up frustrated and disillusioned. However, if you've been looking for a really effective way to create first-rate graphic designs on the Spectrum, and you have the necessary ability, this is the tool you have been waiting for.

Getting it off the screen ...

Of course, even the best graphic designs are not a lot of use in the real world unless you can get them off the monitor screen and onto paper. Fortunately that's a lot less difficult than it sounds – in fact there are several computer printers now on the market that will give you full-colour printouts. It's time to look at just what *is* available on the computer printer scene.

Chapter Eight

Printers and Printer Interfaces

by Phil Gardner

Getting it on paper ...

Way back in the dear dead days when men were men and ZX81s were computers I can remember copying down screen LISTings of my painfully-entered programs by hand as I tried to work out why subroutine 3000 was constantly giving me an error message whenever I jumped to it from 290 ... Yes, I could have scrolled backwards and forwards all the time, but the keys were beginning to get tired and so was I. The fact is that there's no substitute for 'hard copy' – a listing on paper which you can check, recheck, and write rude messages on. And while pen and paper may have been OK for William Shakespeare, BASIC spelling and grammar are a lot less flexible than English, and computers don't take kindly to mistakes. This is probably how many people start thinking about printers, and once you've actually started to use one for your program listings you'll never want to use anything else. And listings are only the beginning. Plenty of programs produce displays and information that are lost as soon as they reach the next CLS command – unless, of course, your printer has made a hard copy for you first.

Maybe you're a games player – and maybe you don't think a printer would be much use to you. But imagine playing a game like *The Hobbit* and being able to keep a printed record of what you'd done and the appalling things that had happened to you as a result. At least, next time you would have a way of knowing what *not* to do. With a good printer you could even run off a copy of the better screen displays – or your own masterpiece, hot off your RD Digital Tracer or your light pen (see Chapter 7)!

Making the choice

There are so many printers on the market at the moment that you may well find it rather hard to sort out what you want and what the best one is for your particular needs. The good news is that the prices – which used to be geared to business users – have fallen rapidly in the last year or two and new models, with more and better facilities, are appearing all the time. This is wonderful, but it makes the choice even more difficult! Still, if you just want it for program listings, the look of what it produces really isn't important (as long as you can read it!) and your best bet might well be Sinclair's much-abused but perfectly serviceable **ZX Printer** if you can get it (production has now ceased, though paper will still be produced). Otherwise, your best bet is the **Dean Alphacom**. It costs £59.95, produces a nice blue printout, and uses rather cheaper paper. Neither of these printers needs a special interface. On the other hand, if you want something a bit more impressive then you'll almost certainly need to think in terms of a *full-width* printer – something that can manage to print 80 letters to a line rather than the 32-letter maximum for the ZX and the Dean.

Full-width printers

There are four main types:

- Daisywheel printers
- Dot-matrix printers
- Ink-jet printers
- Graphic plotters

Daisywheel printers take their rather poetic name from the print element, a circle of moulded letters mounted on radiating spokes like the petals of a daisy. The wheel turns at high speed to place the required letter at the top, over the paper, and a hammer strikes it against the ribbon to print the character. Daisywheels produce beautiful printouts which can match the best electronic typewriters, but they tend to be slow (12–20 characters per second) and noisy. Some can underline, or produce bold type, but unless you actually change the daisywheel (which is possible, of course) you can't have different typefaces on the same page.

Dot-matrix printers have a very different sort of print head – just a row of seven, nine, or sometimes even more tiny wires. As the head

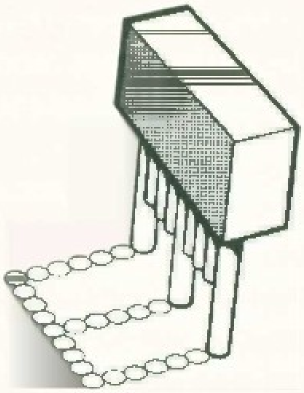

Fig. 8.1. How a letter is formed on a dot-matrix printer.

moves along the paper, one dot-width at a time, some of these wires are pressed against the ribbon (see Fig. 8.1) and a letter is formed out of the dot pattern they make, in rather the same way a letter-shape is built up on your TV screen. 'Thermal' printers work by heating the wires to create dots on heat-sensitive paper. Dot-matrix printers of any kind are cheaper, faster (typically 80 cps) and more versatile than daisywheels. As well as ordinary text they can produce underlining, bold type, and even italic, enlarged, and condensed faces, not to mention superscripts, subscripts, and even, in some cases, your very own character sets (fancy brushing up on your Hebrew?). In graphics mode, they can also copy a screen display, which is how we got started on this topic in the first place. Some daisywheels can, too, but not well!

The only problem with dot-matrix printers is those Reader's Digest personalised letter dots, which are a dead giveaway that you've been using a word processor to type your party invitations, your job application, your homework, or whatever. Some people still frown on this (they seem to frown on anything that helps to make life a bit simpler), though admittedly the cheaper printers give them reason to by truncating the descenders (the 'tails') on letters like y and p. This is especially likely if the printer only uses a small matrix like 5 × 7. Still, even a 9 × 9 matrix gives much better results, and some of the latest machines use matrices up to 12 × 18, and a character set that's almost up to daisywheel standards. Some will even let you print in colour, using a special multi-coloured ribbon (guess how much *that'll* set you back!) but it's a clumsy process – and there's a better way.

Ink-jet printers work, as the name suggests, by squirting little dots of powdered ink from four tiny jets, building up a full-colour image from the four primary printing colours – black, cyan (blue), magenta and yellow. The results are very attractive, though on most types of paper the ink spreads a little, so the image tends to look a bit fuzzy. The price, however, is not so nice – around £500 plus VAT. Maybe the ZX isn't so bad after all ...

Ink-jet printers are really a spinoff from dot matrix printers – they use the same dots of ink but apply it in a different way. The *graphics plotter* actually draws out each letter and graphic by 'writing' it with a pen, which is lifted up and then put down again for the next letter. And colour? It's simple – you just use four pens! Plotters are at their best with graphs and charts; if you're planning to copy the ceiling of the Sistine Chapel, they're not really going to be up to the job, and text looks a bit spidery. Good ones are expensive (both the hardware and the software are fairly complex) and most of them are painfully slow.

Something else that varies a lot from printer to printer is the mechanism used to feed through the paper. There are three basic types, and some printers cater for all three! *Tractor feed* uses pins that engage in holes punched along the edge of a continuous perforated strip of paper and 'pull' it into the printer. *Friction feed* is what's used in most common or garden typewriters – the paper is gripped between rubber rollers. *Roll feed* allows you to place a full roll of paper inside the printer itself; most friction feed printers can handle this as an option.

So which one do you choose? With printers, as with everything else, there are arguments for and against every choice, so to arm you against persistent smooth-talking salesmen we've listed them in Table 8.1. These are the things you need to know to make your choice; the rest is up to you.

Table 8.1. The pros and cons of printers.

Type	Pros	Cons
Ultra-cheap	Very cheap.	Illegible, slow, few facilities.
Dot-matrix	Affordable, fast, good facilities.	Not letter quality. Can be noisy.
Daisywheel	Letter quality.	Noisy (often very!). Usually expensive. Slow and inflexible.
Ink-jet	Good colour.	Very expensive.
Plotters	Quiet. Line graphics.	Fairly expensive. Poor text quality.

Spectrum calling printer, are you receiving me?

Unfortunately, the only printers that plug straight into the Spectrum are the ZX and the more expensive but extremely good Dean Alphacom. For anything else you need an interface. Remember that with a more elaborate printer you've effectively bought a computer that just happens to be a printer as well – it has its own ideas of what number codes from the Spectrum mean, and you need a translator in the middle to avoid confusion. As we saw in Chapter 1, the interface is the combination of hardware and software that does the job. There are only two fundamentally different types of interface:

(a) *serial*, which sends data one bit at a time down one wire;
(b) *parallel*, which sends it one byte at a time, with the eight bits going simultaneously (in parallel) down eight wires.

Only two of the various standard interfaces are worth considering for use with the Spectrum. The *RS232* (or RS232C) is the commonest serial interface, while the *Centronics* (named after the printer manufacturer who developed it) is the commonest parallel interface. Most printers can be fitted with either type, though you often have to pay more (heaven alone knows why) for an RS232.

Making the choice

In practice it doesn't make much difference which interface you choose, though an RS232 might allow you to link up a modem as well (see Chapter 9). Only the computer and the printer can tell the difference. If you have Interface 1, with its built-in RS232 apparently disguised as a joystick socket, you might be tempted to use that. But be warned – Sinclair BASIC can only handle this socket rather clumsily, and you can't just use LPRINT. Besides, you might want to connect it to something else.

The alternative, whether or not you have Interface 1, is an add-on interface from another supplier (see Table 8.2). Most of these are Centronics, though a couple offer both Centronics and RS232 in a single unit. This could be handy if you didn't want to be tied to one or the other – or if you wanted to run two printers! All these interfaces arrive with software either on board or on cassette so that you can set them up to translate LPRINT, LLIST and COPY into language that any printer of reasonable intelligence will understand and act upon. The code for 'pound' (£) from the Spectrum, for

Table 8.2. Printer interfaces.

Manufacturer and name	Type	Software	Price	Documentation	Remarks
Euroelectronics ZX Lprint III	Cen + RS232	ROM	£34.95 (+£9.95 cable)	Brief, fairly clear, assumes some knowledge of control codes, etc.	Compact, well-made, easy to use.
Hilderbay	Cen	Cassette	£45.00	Good, clear.	Good customising facilities. Difficult to fit and remove without using undue force which could damage the PCB. A 'professional' version at £99 includes a printer buffer.
Kempston Centronics Interface E	Cen	ROM	£45.00	Very poor. I found it totally confusing.	Good facilities let down on our sample by a cable fault also reported by some other users. We were not able to transmit the 'ESCAPE' code essential to switch dot-matrix printers to italic and other useful modes.
Morex	Cen + RS232	Cassette	£49.00 (+£12.65 RS232 cable)	Very detailed. Useful.	Easy to fit and easy to use – the software keeps things simple for you and once you have saved it to tape/Microdrive and incorporated it in your Tasword program you can almost forget about it!

instance, produces an open-quote on my printer, and there's no easy way of printing a pound sign. But with the **Hilderbay** interface, and most others with cassette-based software, you can *redefine* the characters so that, for instance, '£' really does print a pound sign.

If you want to use a full-size printer you will almost certainly be using a word processing program – and by far the best for the Spectrum is **Tasword Two**, which we used to write this book. Almost all the available interfaces come with instructions on how to use them with Tasword. If yours doesn't, be suspicious. It's important.

You will also want to copy the screen display. If you're just printing up text, this is easy, but a high-resolution copy of your favourite screen graphic needs a machine-code routine if you don't plan to spend all night waiting for it, and it must be designed to work with your printer. Make sure your interface provides this facility, preferably in ROM but at least on cassette, and that it'll work with your chosen printer.

The final ingredient is a cable, which ought to be supplied with the interface. Most actually come as an integral part of it. The **Euro Electronics Lprint III** has detachable cables, Centronics or RS232 as required. The **Morex** is the same. The plug on the cable should fit any standard Centronics or RS232 socket (see Figs. 8.2 and 8.3) on your printer – but do check before you buy.

We can fix you up, guv ...

I have heard stories about people trying to link their computer to an old teleprinter – they're obsolete, so they're cheap to get hold of. But unless you're a demon with a soldering iron you'd be well advised to forget this idea. Even if it worked, the resulting system would be bulky, noisy, slow, and thoroughly unsatisfactory. A more sensible idea might be to adapt an electronic typewriter, especially if you have one already. Check with your dealer that the necessary modification is possible, and check the price, too, because you could be in for a shock. The cost of printers has come down so much that it

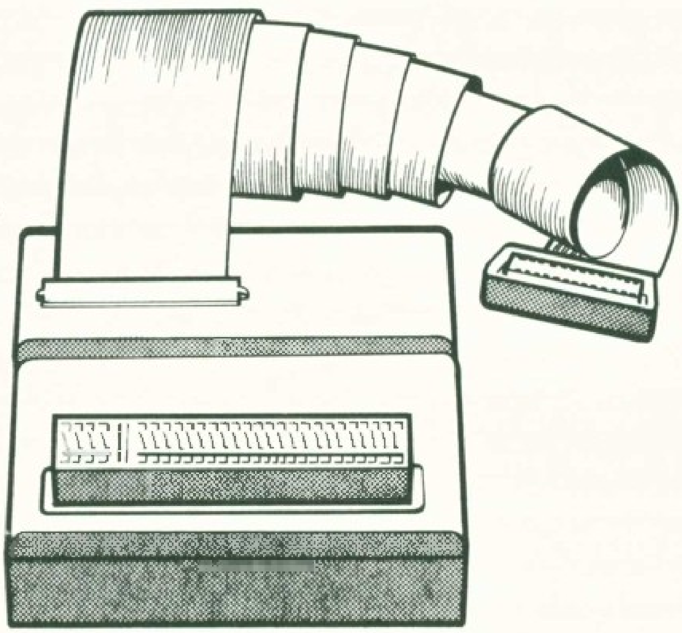

Fig. 8.2. A Centronics interface.

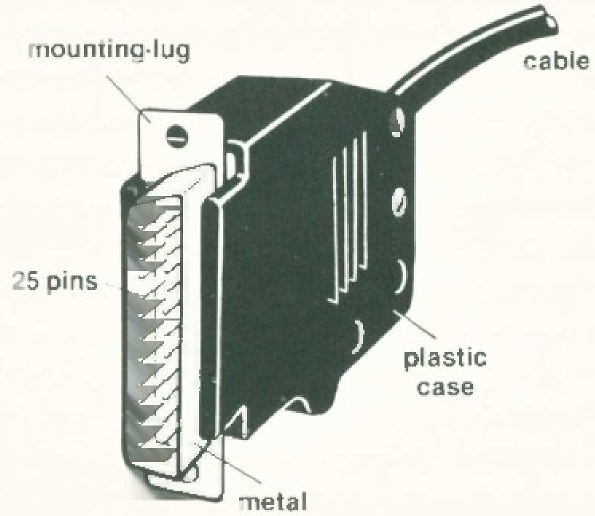

Fig. 8.3. A standard RS232 interface connector. This end connects to the printer.

might hardly be worth it, and you can always keep the typewriter as a back-up! Until quite recently Hilderbay sold a ready-modified version of the Olivetti Praxis, but this is being discontinued. All the same, they may well be able to advise you.

Some of the more recent electronic daisywheel typewriters are designed to act as printers: you simply buy the necessary interface as an extra and plug it in. If you basically want a good typewriter, which from time to time you could use as a printer, this might just be a worthwhile option – though not a cheap one. You can buy the **Silver-Reed EX43** or **EX44** for just under £300, but it will cost you at least £100 more for a parallel interface and £45 for the Spectrum interface! And after all that you'll be left with a slow (5–6 cps) unidirectional printer with no buffer, no bold, and no underline. The **Brother CE51** (£300) and **CE60** (£430) are faster and have more facilities, but the interface here – admittedly a dual parallel/serial one – costs a staggering £200. For the total money you'll be spending you could buy a very good daisywheel printer – and if you really want a typewriter then a simple ten-line BASIC program will make it behave like one by printing out everything you type on the Spectrum keyboard.

The choice is yours

You've decided what sort of printer you want. You know how to connect it to your Spectrum. You know how to get it working. But *which one do you choose?* To help you, here's a point-by-point guide to trying and buying your printer – and we'll round off with a quick look at some of the models currently on sale. Again, this doesn't pretend to be complete (there are just too many printers!) so check prices and other details in an up-to-date magazine before you decide to buy.

- *Try before you buy.* Don't just rely on the ads and the brochures – read a magazine review first, then go to a dealer and (ideally) see the printer being driven from a Spectrum. At least use the 'self-test' facility. See if you agree or disagree with the magazine review, and check out any problems or disadvantages it mentions.

- *Does it have the facilities you want?* If it's daisywheel, do you like the available faces? If it's dot-matrix, can you get all the different typefaces you want? Can you load different character sets into it? Does it have a graphics (or bit-image) mode to let you copy pictures from screen?

- *Speed and noise.* Remember that quoted speeds (such as 140 cps) are maximum speeds – they don't allow for things like carriage return, or the time taken for a line-feed (i.e. moving the paper up by one line). And remember that if the printer offers 'near letter quality' it's going to be a lot slower in that mode. See how long it takes to print a full page of text – and listen to the noise. You're the one who has to live with it!

- *Back-up and service.* If the printer is made by a large and reputable manufacturer, this shouldn't be a problem. But if it's made, or imported, by a small or obscure company, make quite sure that the service arrangements are up to scratch. This is another good reason to buy from local dealers rather than mail order.

- *Ribbons and paper.* How easily can you get new ribbons? And how much will they cost? Some printers use typewriter ribbons (cheap!). Others are very pricey – if they're carbon ribbons, ask if you can use them more than once. Paper is readily available. The commonest sort for most full-width printers with tractor feed is 8" × 11" continuous (fanfold) paper. You can tear off individual sheets along the perforations, and also the strips at the sides that carry the sprocket holes for the tractor feed mechanism. A box of 2000 sheets should cost about £10–£14. Don't buy the stuff with green lines on it unless you find it irresistible (does anyone?). If the printer you're looking at will only take special paper, consider its cost, availability, and appearance.

- *Type style.* Does it look good? Do *you* like it? Do you like the other options (if any)? If it's dot-matrix, just how dotty is it? Are the characters well-designed, with proper descenders?

- *Price.* You know roughly how much you can afford. Is the printer you're considering really good value for money compared with other models in the same price range? Check out the magazine reviews. Can you get it more cheaply somewhere else, by mail order, for instance? You may be able to beat down the dealer's price!

Make yourself a checklist based on these points, and any others that may be important to you (strength, size, whether it'll match the new curtains ...).

Table 8.3. A selection of printers.

Name	Matrix	Feed	Chars per line	Speed in cps	Extra for RS232?	Price
(1) *Dot-matrix printers*						
Epson RX-80 FT	9 × 9	T/F/R	80	100	No	£280
Epson FX-80	9 × 9	T/F/R	80/137	160	Yes	£380
Mannesmann Tally MT80	9 × 8	T/F	80/142	80	Yes	£225
Microline 80	5 × 7	T/F/R	80/132	80	Yes	£220
Microline 82A	9 × 7	T/F/R	80/132	120	Yes	£315
Panther DX109	9 × 9	T/F	80/158	80	Yes	£330
Seikosha GP50S	5 × 8	F/R	46	40		£140
Seikosha GP500S	5 × 7	T	80	50		£230
Shinwa CTI CP80	9 × 13	T/F	80/142	80	Yes	£240
Star Delta-10	9 × 9	T/F	80/136	160	Yes	£360
Star Gemini-10X	9 × 9	T/F/R	80/132	120	Yes	£230
Taxan/Kaga KP810	9 × 11	T/F/R	80/132	140	Yes	£345
Walters WM-80	7 × 8	T/F	80/142	80	Yes	£240
(2) *Daisywheel printers*						
Brother HR-15		T/F	165	13	No	£395
Juki 6100		F (T)	165	18	Yes	£375
Silver Reed EXP500		F		14	Yes	£299
Smith-Corona TP1		F (T)	126	12	No	£245
(3) *Others*						
Integrex ink-jet		F/R	1280 dots/line	37	Yes	£575
MCP 40 Plotter		R	40/80	12	—	£120

Table 8.3 shows a selection of dot-matrix, daisywheel and other printers. Virtually all dot-matrix printers take paper 8.5 inches wide, with a maximum width up to 10 inches. The daisywheel printers shown can all take paper up to 13 inches wide. The letters T, F and R stand for traction, friction, and roll. The number of characters per line are quoted for normal and condensed modes, and speeds are those quoted by the manufacturer.

The **Epson RX-80 FT** offers underline, bold, block graphics, high-res graphics, and is bi-directional logic seeking. It's well worth considering – Epson have an excellent reputation. The **Epson FX-80** also offers proportional spacing, plus extra speed and graphics modes, and an optional tractor feed that increases the range of paper widths the printer can handle. It also features very finely programmable line spacing. The RX-80 FT, apart from its slower speed, is virtually the same machine – and therefore excellent value.

The **Mannesmann Tally** offers compressed, double-width, emphasised and underline facilities as well as high-res graphics and bi-directional logic seeking. It's beautifully cased, and good value, but offers no italics. However, there is an optional sound reduction kit and buffer. The **Microline 80** is cheap but has very few facilities for the price; the **Microline 82A** is faster and has underline, block graphics, and bi-directional logic seeking. The **Panther DX109** has a viewdata character set with underline, bold, high-res graphics, and bi-directional logic seeking, but there is no italic and no alternative character sets. It's difficult to recommend unless you want this particular character set.

Seikosha now offer two machines with special Spectrum interfaces already fitted. The **Seikosha GP50S** will only handle paper 5 inches wide, but gives you a choice of seven ribbon colours and produces better print quality than most of this manufacturer's range. The **Seikosha GP500S** only offers bold and high-res graphics options: print appearance is poor, and its lack of facilities makes it rather poor value. It's also extremely noisy. It has a self-inking ribbon and replaceable ink reservoir, but is really little improvement on the earlier **GP100A** machine.

The **Shinwa CTI CP80** has underline, bold, block graphics, high-res graphics and bi-directional logic seeking: good facilities at a good price. The **Star Delta-10** has downloadable character sets, underline, bold, proportional spacing, block graphics, high-res graphics and bi-directional logic seeking – good facilities at a good price. The **Gemini-10X** has underline, bold, block and high-res graphics, and bi-directional logic-seeking – it's fast, with good facilities, and good value for money, besides being cheap to run: it uses ordinary typewriter ribbons! The **Taxan/Kaga KP810** has underline, bold, near letter quality, proportional spacing, block and high-res graphics, downloadable character sets and bi-directional logic seeking. It's a superb machine at a very good price. The **Walters WM80** has underline, bold, 7 international character sets, block and high-res graphics, and bi-directional logic seeking – good value, though it is not especially fast, and has no italics.

Among the daisywheels, the **Brother HR-15** and the **Juki 6100** both offer underline, bold, proportional spacing and bi-directional logic seeking. The Brother is a slightly slower machine. The **Silver Reed EXP500** offers underline, bold, block graphics, and bi-directional in-line mode – a solid machine, and good value. The **Smith-Corona TP1** has only underline and bold as options, but is still excellent value, although it is rather noisy and has fewer characters on its daisywheel than some.

The **Integrex Colorjet** is far from cheap, but probably the best ink-jet printer for home computers. It is available at a lower price, but without the screen dump software, under the Tandy label. The **MCP 40** plotter offers highly flexible reproduction of graphics and screen displays, but the paper is only 4.5 inches wide.

Chapter Nine
Modems and Networking

This chapter is dedicated to putting your computer in touch with other computers – either in the same room (using the networking facility of Interface 1), or elsewhere in the country (using the telephone).

Have you heard the (true) story about the little old lady in Scotland whose phone kept ringing at one o'clock in the morning? When she answered it, all she could hear was a series of whistles and beeps. She used to hang up, because she thought it was an obscene caller. In fact it was a computer trying to reach another computer – and getting a wrong number.

If you thought that sort of thing could only happen to big mainframe computers, then you're in for a surprise. It is perfectly possible to do exactly the same thing with your Spectrum (though you're not quite so likely to get a wrong number). You can 'talk' not only to other Spectrum owners, but to home computer owners throughout the UK, to mainframe computers with massive databases all round the country, and even (theoretically, at least) to similar database computers overseas.

Why your computer should be on the phone

At first sight none of this looks terribly interesting to someone who's just bought a computer. What, after all, is the point of talking to other computers when you've only just figured out how to program the thing?

One of the first really attractive answers to that question came from an organisation called *Micronet 800*, and in their case the magic words were 'free programs'. The result was a genuine interest and a fast-growing subscription list that soon made other commercial enterprises realise the value of this kind of system.

Micronet 800 doesn't just offer free programs, of course – it also offers up-to-date information on the entire microcomputer scene (some of which found its way into the final draft of this book!) and a 'mailbox' facility that allows you to leave messages for fellow-users or collect messages from them. The total runs to some 40000 pages.

However, Micronet 800 is really a sort of subscription-only section of Prestel, British Telecom's own telephone data service: the jargon word for this kind of service, incidentally, is *viewdata*. Neither service is free – apart from the usual charge for using the telephone (normally billed as a local call unless you live somewhere way out in the wilds) there is a £4.30 monthly subscription for Micronet and another £1.69 a month for use of Prestel. Frankly you'll need both if you decide to join, because much of the really interesting information the system has to offer is on the truly massive Prestel database.

Once you've got through to Prestel (of which more in a moment), you are presented with a series of menu pages rather like the menu pages used for complex programs. Each choice you make (using the ordinary number keys on the computer, with a * to indicate the start of the number and an ENTER command to finish it) brings up a more specialised menu, until you reach the specific area of information you were looking for. There's a great deal of help available from the system itself if you miskey a number or get lost, and in any case you'll have a special Prestel directory to help you find your way around.

A very large number of Prestel pages can be accessed for nothing (apart, of course, from the charges mentioned above). However, quite a few of the more specialist pages are run either by closed user groups (who charge, like Micronet, their own membership fees) or by commercial companies, who may expect you to pay up to 50p a page to see their work. You'll get plenty of warning if a page is chargeable, and if you decide to go ahead the charge will simply appear on your next telephone bill. However, if you really want to pour your money away you can also, quite often, order goods or services on Prestel; you can even run a bank account on Prestel if you want to. Normally the supplier or bank will provide a sort of viewdata form which you can fill in from your own computer with the details of what you want. If you find yourself involved with systems of this kind, however, be careful. It's all too easy to run up an enormous bill without realising what you're spending!

The least we can do is talk to each other . . .

If Prestel and Micronet hold no lure for you, then it's worth considering another service that only the phone can give you – data transfer between your computer and someone else's. Until quite recently this was very difficult on the Spectrum, but in the early months of 1984 the market was suddenly flooded with new devices and software to make this possible. As I said in the Introduction, this book was actually edited by phone – I sent completed chapters, in the form of machine code files, down the line to Phil Gardner in Leeds, and he was able to copy them directly onto Microdrive. In fact, very little of this book existed as hard copy (i.e. as text printed on paper) until it was almost ready to be sent to the publishers.

Of course, there are a good many other things you can do with this kind of communication. You can send BASIC programs, prepared messages, machine code routines, screen strings, anything the computer's internal memory is capable of storing. You can type messages to each other on the keyboard, and those messages will appear letter by letter on the receiving computer's screen display. (This may sound slow, but it's extremely useful when a crackle on the line has interrupted your program transmission.) Telephone data transmission is a lot safer and a lot faster than sending cassettes or Microdrive cartridges through the post, and if you're careful it needn't cost a lot more. So how is it all done?

Introducing the modem

The secret behind this particular communications revolution is the *modem*. A modem (short for *mod*ulator and *dem*odulator) is really just another kind of interface, one that happens to specialise in sending and receiving messages. If the theory behind the modem holds no charms for you at all, I have some good news for you. The Prism VTX 5000 modem which I use does everything for you: once you've rung up the computer, switched on, and entered your personal code you can run happily through page after page of Prestel until you get bored. You can set up mailbox messages, send programs and data to your friends, and load up the free programs, without ever worrying about how it's all done. But if your curiosity gets the better of you, this section should help you to understand a few of the basic principles.

When you send a business letter you normally start with a formal opening (Dear Fred, Dear Sir, Sir, depending on how late his payment is) and finish with a standard close (All the best, Yours sincerely, Yours very truly, Our solicitors will be in touch with you tomorrow). For most business letters you will also expect some sort of reply: for instance if you've sent a cheque for some goods you'll expect a delivery note and a receipt with the goods when they arrive. A computer expects these things, too, and the modem's task is to make sure it gets them.

'Dear Z80A ...'

In the case of your computer, the 'message' is likely to take the form of data bytes. As far as another computer is concerned, each of these data bytes is a message in its own right, so each of them needs an 'opening', a 'close', and some sort of reply.

The 'opening' is known as a *start bit*, which can, not surprisingly, be either 0 or 1. The same goes for the 'close', or *stop bit*, at the end of every byte transmitted. Obviously, if one byte has just finished, the computer should be ready to expect the next byte! However, virtually all systems will make a 'reply' to each individual byte, by transmitting a copy of what has been received back to the computer sending the message so that its accuracy can be checked, the reply comes in on a different frequency, so the transmitting computer isn't hung up while it waits for an answer. The jargon term for all this is *full duplex*, and with their usual flair for fouling up the English language the high priests of computing have decreed that the opposite of full duplex is *half duplex*.

That's not quite all. The receiving computer also needs a way of checking that the information it's getting hasn't been corrupted – by a bad line, for instance. (Yes, even computers have trouble with telephone lines – after all, they can dial wrong numbers.) The trick here is to add an extra bit before the stop bit so that the decimal value of the entire code always comes out even (or odd, if the receiving system is expecting odd-numbered bytes). This extra bit is known, logically enough, as a *parity bit*.

How it works

Obviously you can't send more than one bit of information at a time down the phone line – you need to break individual bytes of data up

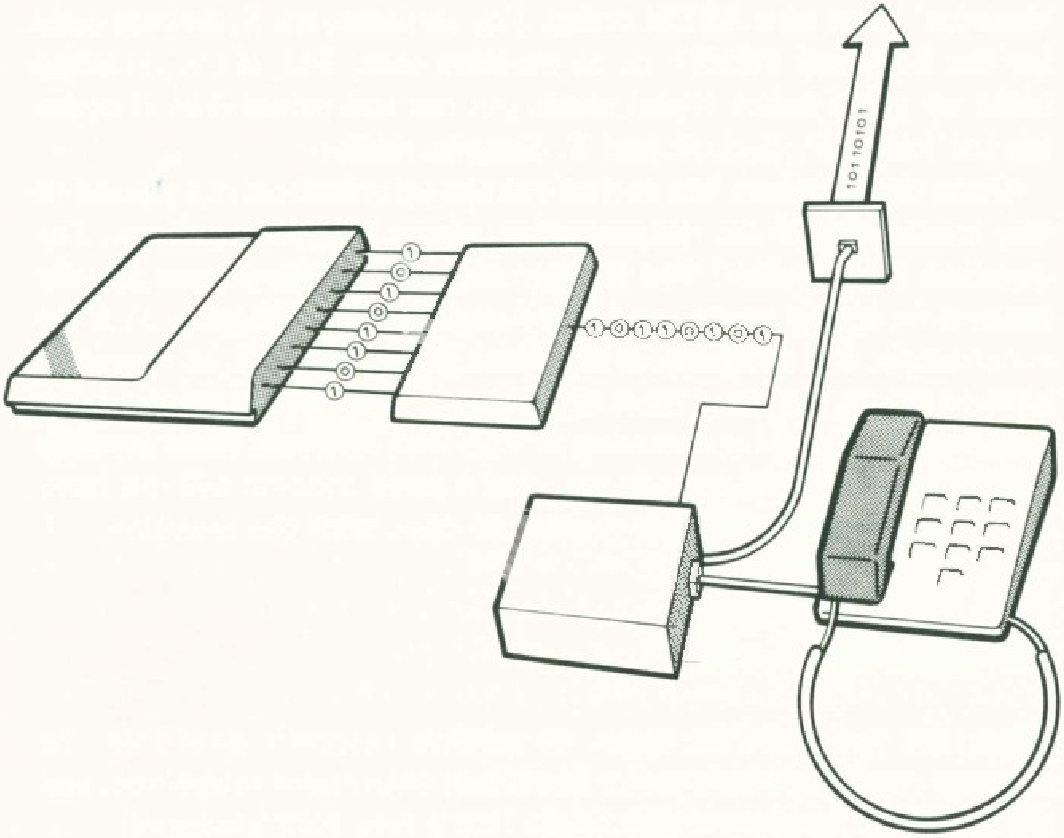

Fig. 9.1. How modems work. The parallel byte from the computer (left) is broken into serial bits by the RS232 interface. These are transmitted along the telephone line by the modem.

into their component bits before you send them. Usually this is done by an RS232 serial interface (see Fig. 9.1). Most UK systems will not accept a full 8 bits of data; they'll read the last bit as a parity bit. This sounds illogical, but in fact it doesn't actually cause any problems. It simply means that the computer can only send out bytes with a decimal value of 127 or less. And believe it or not, it doesn't matter.

If you're now feeling rather confused, take a quick look back at Chapter 1, where I explained how signals from your keyboard, including single-entry keywords, are read as numbers by the CPU. These numbers are pretty much standardised on all computers, and are known as *ASCII* codes. (ASCII stands for American Standard for the Coding and Interchange of Information.) However, ASCII codes only cover letters, numbers and symbols, not the Spectrum's own token system for understanding keywords. That means that ASCII code numbers don't need to go any higher than 127, which is useful if you want your Spectrum to talk to an entirely different computer! As for the tokens, the numbers above 127, the interface 'decodes' them into ASCII codes. For instance, the Spectrum code for the keyword PRINT is, as we have seen, 245. If the interface picks up a 245 token it just sends out the individual ASCII codes for the letters P, R, I, N and T. Simple, really.

Once the interface has sorted out the information, it's passed to the modem itself for transmission along the phone lines. The question is, how fast? The answer depends on the precise job the modem is trying to do, and the *baud rate* (see Chapter 5) is set accordingly, but more about this later. Normally you, as operator, have to set up the baud rates in advance – one rate for transmitting data, and a second for receiving it. This isn't difficult; there are only a few standard baud rates in use, and a good modem will just have a dial or a simple software routine to cope with the selection.

Choosing your modem

There are two different types of modem currently available: the *acoustic coupler*, which looks like a sort of inverted telephone, and the *hardwire modem*, which looks like a box with knobs on.

Acoustic couplers are designed to fit over the earpieces and mouthpieces of your telephone: a speaker sends the data as audible tones through the microphone in the mouthpiece, and a microphone picks up transmitted data from the speaker in the earpiece. You simply connect the computer to the coupler and send your data directly down the telephone. Acoustic couplers have several advantages. They're light, and the portable ones are easy to carry around with you. They don't make any electrical connection to the telephone system, so there's no risk of a short circuit, which is the kind of thing that tends to make British Telecom a bit nervous. The snag is that background noise can sometimes creep through the rubber connectors and turn up as 'garbage' on the computer at the other end. And to avoid this, as far as possible, you'll need to take some trouble about setting up the connection in the first place. Then, of course, there's the simple problem of talking – you can hardly use the phone for a conversation when there's an acoustic coupler fitted to it, so you'll have to keep taking it on and off.

The so-called hardwire modem *does* make British Telecom nervous, because it's usually connected directly into the telephone system with a standard BT plug and socket. Modems of this kind must have Telecom approval before you can connect them in this way. (It's not illegal to *own* one, but this is small comfort!) The point is, of course, that the part of the modem needing electrical power must be completely isolated from the part connected to the phone system. Some suppliers cloud the issue by talking about 'approved components'. Unless the entire unit is approved, you'd better forget it. It's a pity that good old BT take so long to approve new products – they are trying to cut down the time it takes, but in a fast-moving industry like this one, three months is about three months too long.

Aside from this, there are three main factors that will influence your choice of modem:

- What you want to use it for.
- What features are important to you.
- How much you are willing to pay.
- How easy it will be to connect the modem to your Spectrum.

If you only want to use a modem to swap programs and data with your friends, then there's not much point in buying something that will get you through to every viewdata system on the continent of North America. On the other hand, if you need worldwide database access then something like the MicroMyte 160 IQ/D is not going to do you much good.

Do you want the computer to do the dialling for you? It's possible – at a price – but you're going to pay through the nose for it, so make sure it's that important. Some units offer this as a plug-in extra, so you might be well advised to wait and see how useful it would really be. Some will also 'answer the phone' to another computer!

If you have an RS232 interface (for instance, one of those supplied as an option on some printer interfaces) then virtually the whole range of currently available modems is open to you – in theory. However, first you must make sure that the Spectrum can actually work with the modem you want (modems, like printers, need to speak the same language as your computer). Then you must make sure that the interface you have is suitable (sadly the one in Interface 1 isn't likely to be). Finally, you need someone to wire up the connections for you. This is not a job for the amateur electrician, as the supposedly standard RS232 connections can vary wildly, so make sure the job is done for you by someone who knows his business.

Bearing all these points in mind, you can now take a look at what the market currently has to offer – but please remember that by the time you read this the choice could be even wider, and many prices may have fallen.

The Prism VTX 5000

For the Spectrum, the **Prism VTX 5000 modem** is indisputably the market leader. It is made by O.E. Ltd, in Cumbria, and marketed by Prism. The basic unit uses on-board firmware to get the Spectrum in full communication with viewdata. The idea is to leave the modem connected to your computer and to the phone socket all the time – your telephone plugs into a socket in the modem itself. As soon as you power up you get a Micronet screen display – pressing ENTER gives you a simple menu which includes the option to return to BASIC. You can also set up a message for transmission, log on or off Prestel/Micronet, or prepare to load a program.

Controls are few and simple – just an on/off line switch and a three-position switch that lets you set the modem up for receiving or transmitting data, or for Micronet. To use the modem on Prestel, just enter your personal code number, ring up the computer, turn on the line switch, and hang up your phone (yes, really!) The Prestel logo will appear on screen, and you'll be invited to enter a pass code consisting of four numbers. Do that, and you're in – the rest is up to you.

The first and biggest surprise is to find your trusty Spectrum apparently producing a 40-column display. In fact this has more to do with the firmware on board the VTX 5000, but it's decidedly unsettling! Once you've got over this initial shock you can look at any messages that have been left for you (the system will tell you they are there), leave prepared messages for other people, and then go off in search of free programs, information, or anything else you fancy.

For direct communication with another computer you need an extra piece of software, which arrives as a BASIC loader program and a machine code routine. I transferred mine to Microdrive straight away. Using this program you can send prepared messages directly to your correspondent, or transmit BASIC and machine code programs loaded from tape or Microdrive. We found from experience that any machine code routine longer than about 2K tended to be corrupted in transmission, but there's nothing to stop you sending even quite a long file in 2K chunks.

The baud rates for the Prism can't be changed – they're fixed on one of the Prestel standards, which is 1200 for receiving information and a painfully slow 75 for sending it. This prevents you from getting into some so-called 'bulletin board' services, which use a 300/300 baud standard. Incidentally, the 300/300 standard will also give you full access to all Prestel pages. Other than that the Prism has no

Table 9.1. Modem comparison chart. (Remember that any modem requiring an RS232 interface will cost you £40–£50 more than the quoted price unless you have an RS232 already. Both 1200/75 and 300/300 are standard Prestel baud rates – 300/300 actually gives you access to more services.)

Name	Type	Baud rate and duplex	Needs RS232?	Telephone answering?	Price
Concord V22	Hardwire	1200 Full 600 300	Yes	Yes (can be switched off)	£547
Dacom Buzzbox	Hardwire	300/300 Full	Yes	No	£69.95
Maplin	Hardwire	300/300 Full	Yes	No	£39.95 (kit only)
MicroMyte 160 IQ/D	Acoustic	1032 Non-standard	No	No	£99
Minor Miracles WS2000	Hardwire	300/300 Full 1200/75 Full 1200 Half	Yes	Yes as extra for £39	£115 (awaiting BT approval)
Pace Grapevine	Hardwire	300/300 Full 1200/75 Full	Yes	No	£125 (awaiting BT approval)
Prism VTX 5000	Hardwire	1200/75 Full	No	No	£100
Tandata	Hardwire	1200/75 Full	Yes	No	£99

disadvantages, and at £100 it's definitely a sound investment. All the same, it has competition, so I have included a brief point-by-point guide to the rest of the market; the technical details are summarised in Table 9.1.

The **Concord V22 Data Modem** is a Rolls-Royce model – it will even cope with distortion on the phone line, and when answering a call from another computer it will automatically switch to the right baud rate. The **Dacom Buzzbox** has mains and battery-powered versions available; the mains is cheapest at £69.95. It's a very economical choice if you need an RS232 interface anyway, but not otherwise. The **Maplin modem** is the cheapest hardwire modem on the market, but only buy it if you're willing and able to assemble it yourself, because it's a kit. The **MicroMyte 160 IQ/D** is an acoustic coupler for £99 – but this one uses the cassette port, not an RS232C, so there are no hidden extras. Unfortunately, it's light on features, and is fixed on a totally non-standard baud rate of 1032. This means that you can only use it to communicate with other MicroMytes. The **Minor Miracles WS2000 World Modem** is not yet BT-approved, but has a massive range of features, coping with European (CCITT) and American (Bell) standards. The **Pace Grapevine** is not yet BT-approved, either. It can cope with European and American transmission standards – there's also a self-test mode. No standard connection for the Spectrum exists yet, but ask about one if you're interested. The **Tandata TM100**, like the Prism, runs only on the 1200/75 baud full duplex standard, and costs, like the Prism, around £100. Unlike the Prism, you need an RS232C interface for it. However, there are extras – in this case an ability to store up to eight phone numbers, each up to 16 digits long, each with its own password and log-on code. You can even change your own identity code and passwords.

Putting your Spectrum on TV

If Prestel doesn't appeal, there is another source of free programs on tap, almost all day and every day. If you can call up the BBC's Ceefax pages on your television, then you may already know about 'Telesoftware'. The **TTX2000**, another product from the award-winning Cumbrian team at O.E. Ltd, can actually pick this software up and load it into your computer. Its provisional £125 asking price compares favourably with the £225 asking price for Acorn's adaptor for the BBC Micro. At the time of going to press full details were not available, and the software was not ready. The production model should have this on board, and like the VTX 5000 this unit will go straight into action when you power up the system (the jargon term is *automatic boot*). It will include four channel tuners, and will provide a full colour display and, again like the VTX 5000, a 40-column display.

Net service

You only really need a modem if the computer you want to talk to is at the other end of a telephone. If it's in the same room, then Interface 1 can provide an even better link, with a speed of operation that will leave you breathless. If you have two Spectrums, two Interface 1s, and two TVs (not all that uncommon these days) connect them up and start by trying out the 'net game' listed in the manual. If you have Microdrive, too, you'll find this game on the demonstration cartridge. The first time I LOADed it and transferred it to the other computer on the network everything happened so quickly that I missed it completely – suddenly, there it was on the other screen!

As with Microdrive, there are four main methods of swapping information on the network:

- With standard LOAD and SAVE commands (e.g. LOAD*"n";2) – program names aren't used on the network
- By using INPUT and PRINT
- By using INKEY$
- By using the MOVE command

The last three methods deserve a closer examination.

INPUT and PRINT

If you take another look at the Microdrive commands in Chapter 5 you'll find that most of them, with minor modifications, will operate between two networked computers. The difference is that you must program *both* computers. The simplest way to describe it is to start with an example. This one uses the 'rooms' Microdrive file set up in Chapter 5, and incidentally demonstrates just how useful the new Microdrive commands can be:

```
110 FORMAT "n";1
120 OPEN #6:"m";1;"rooms"
130 OPEN #7:"n";2
140 FOR j=1 TO 4
150 INPUT #6;r$
160 PRINT #7;r$
170 NEXT j
180 CLOSE #6: CLOSE #7
```

As you can see, the point of this program is to read the Microdrive file and copy its contents to another computer. However, there are a few familiar commands operating in an unfamiliar way. Line 110, for instance, uses the FORMAT command to give the computer a 'station number'. This is necessary so that other computers (there could be as many as 63!) can send messages and data specifically to you. In fact you don't really need to FORMAT station 1 – any Spectrum fitted with Interface 1 will automatically assume that it's station 1 until you tell it something different. Just the same, FORMATting on the net is a good habit to get into. Incidentally, it doesn't wipe the computer, or affect the resident program in any way.

Line 120 opens the Microdrive file for reading in the usual way, but this time line 130 creates a stream to a network channel. In this case the channel is the computer with station number 2, and another buffer is set up to store this information. The FOR... NEXT loop loads the room descriptions one by one into the computer, and line 160 PRINTS them into the network buffer. When both streams are CLOSEd in line 180, the network buffer is ready to be loaded into the computer with station number 2.

Of course, that's only half the story. Station 2 needs some programming, too, or you will just be left with an ugly black border around your screen display and the feeling that nothing is ever going to happen again. In this case the program for station number 2 might look like this:

```
10 FORMAT "n";2
20 DIM r$(4,20)
30 OPEN #4;"n";1
40 FOR j=1 TO 4
50 INPUT #4;r$(j)
60 PRINT r$(j)
70 NEXT j
80 CLOSE #4
```

Notice the 'mirror-image' effect here. In line 10 the computer is FORMATted as station 2. Line 20 sets the DIMensions of the array, much as we did in the earlier Microdrive program in Chapter 5. Line 30 OPENs stream 4 to the network channel from computer 1, and sets up a buffer to deal with the information coming in. The FOR...NEXT loop reads the data into the r$ array and PRINTs it on screen to give you visual confirmation. Finally, line 80 CLOSEs the stream and frees the memory space occupied by the buffer. Incidentally, it also frees station 1, whose operator is probably getting a bit bored with the black border round his screen by now.

One snag of network operations in general is that when two Spectrums are linked on it, only one at a time can actually do anything useful.

INKEY$

Confusingly enough, INKEY$ can be used to get hold of the first character in a buffer that may have nothing at all to do with the keyboard. For instance, if you program computer 2 with

```
10 OPEN #12:"n";1
20 PRINT INKEY$#12;
30 CLOSE #12: GO TO 10
```

and computer 1 with

```
10 OPEN #11:"n";2
20 IF INKEY$<>"" THEN  GO TO 20

30 PAUSE 10
40 PRINT #11;INKEY$;#2;INKEY$;
50 CLOSE #11: GO TO 10
```

then computer 1's operator can have the decidedly odd experience of typing keyboard input onto the screen display of computer 2 and onto his own at the same time. Notice once again that computer 2 is effectively 'locked up' in the loop this program creates. Incidentally, don't try entering spaces on computer 1 or you'll BREAK into the program! If you want to space your words, use asterisks! You can't use the DELETE or cursor keys, either.

When all this ceases to fascinate you, try something more useful. Alter line 30 of computer 2's program to

```
30 GO TO 20
```

and NEW computer 1. Now enter this program into computer 1:

```
  10 OPEN #11;"n";2
  20 PRINT #11;"This is a messag
e for computer 2"
  30 CLOSE #11
```

RUN it, and then RUN the program in computer 2. Result? Your message will appear in full on the other screen, followed by the message 'End of file'. Of course, there's nothing to stop you sending entire screenfuls of information this way – especially useful for micro users in education.

MOVE

As noted in Chapter 5, the MOVE command can only be used with certain types of file. However, if you still have the Microdrive file 'rooms' handy, you might like to try it out on the network. Program computer 1 with

```
10 OPEN #7;"n";2
20 MOVE "m";1;"rooms" TO #7
30 CLOSE #7
```

Now program computer 2 with

```
10 FORMAT "n";2
20 OPEN #7;"n";1
30 MOVE "n";1 TO #2
40 CLOSE #7
```

Now RUN both programs – and the content of the 'rooms' file will appear almost instantly on computer 2's screen display. You could equally well have transferred the file to computer 2's resident Microdrive, or to its ZX or ZX compatible printer. Now try altering the program in computer 2 as follows:

```
30 DIM r$(4,20)
40 FOR j=1 TO 4
50 INPUT #7;r$(j): PRINT r$(j)
60 NEXT j
70 CLOSE #7
```

RUN both programs again. This time, as well as getting a screen display, computer 2 has the data file in the string array r$. Try typing in PRINT r$(1) and you'll see this for yourself. It's just one more example of how powerful this command can be.

Using the net

You can get some idea of the possibilities on the net from the Spectrum Adventure in Appendix 1, which includes a networking option for two computers. Each player has to wait for the other one to move but, as you will see, it's perfectly possible for data items to be 'swapped over' from one computer to the other and used in programs. Using the INPUT and PRINT commands, it's very easy to type in a variable that is labelled, say, 'loc', in computer 1's program, and read it into computer 2's as 'loc2'. This means, as in the Adventure, that both computers can be running the same program while making a useful information exchange.

There's only been space and time here to explore the possibilities of linking two computers. But you *can* link up to 64! (Mind you, don't link the first and last into a loop, or Sinclair Research have hinted that dreadful things will happen.) I've seen at least one published adventure game for such a set-up. Are there no limits to the net?

Chapter Ten
TVs and Monitors
by Mike Scott Rohan

The Spectrum may have been designed to work with ordinary TVs, but its display colour can be pretty hard on the eyes after an hour or so. It's nice that you don't have to pay extra to get full-colour graphics from your Spectrum, but take a close look at that picture. The individual dots that make up the display seem to be constantly changing colour at the edges, sending regular ripples and shimmers through the images. This, appropriately enough, is known as *dot crawl*. Even the finest tuning doesn't help. If it's not the neighbour's motorbike, what's the problem?

Round and round the circuit board ...

Most TV sets can only receive signals through the aerial socket – and the only signals they're equipped to deal with are broadcast signals from a TV station. These signals are *modulated* – the TV circuitry has to *demodulate* them to sort out the signals it needs from the 'carrier wave' that allowed them to reach the TV in the first place.

Because it uses the aerial socket, the Spectrum has to do some pretty strange things to its video signal in order to send it to the TV – in fact, it has to *modulate* the signal which is then demodulated by the TV! The unmodulated signal inside the Spectrum is much closer to what the tube electronics can cope with – and like good French novels, it loses a lot in the translation. It works, but it's not efficient. You can get an idea of the difference by comparing the Teletext picture produced by a properly equipped Teletext TV set with the broadcast version of the same picture often shown on the air outside normal programme hours. The Oracle or Ceefax pictures generated in the set itself always look clearer, steadier, and stronger.

The rather watery TV display is entirely a result of the roundabout route the signal has to take to get there. Every stage adds its own package of interference. A *monitor* is designed to take a short cut. Because it has special inputs and circuitry, it can take the signal out of the computer at a much earlier stage, and lead it directly to the video circuits. At one stage it can be taken out as a single signal containing all the colours in the image, but not yet modulated for broadcast – a *composite* signal – and at an earlier stage as three separate signals containing the red, green and blue elements of the pictures (an RGB signal). A few home computers, such as the BBC, offer all three forms (broadcast, composite, and RGB) – but the Spectrum only seems to offer broadcast. Does that mean you can't use a monitor at all? The answer is 'no' – it's perfectly possible, using the existing circuitry, but it just hasn't been done.

Monitor output the hard way

The Spectrum uses a fairly standard design of microchip to produce its video signals, and this has extra terminals to produce a composite signal. It seems the Spectrum's original designers didn't think anyone would want monitor-quality output, and so – keeping costs down, again – they didn't provide a proper output socket. However, the signals can easily be taken out via the user port at the back, as long as you don't mind sacrificing your guarantee, and as long as you know someone who's got the necessary experience of computer electronics. This kind of work isn't easy, especially when it involves soldering, and one small slip could leave you with about a crumpled fiver's worth of junk components. If you want to risk that, fine – but please don't complain to us.

On Issue 1 and 2 machines the VID strap will have to be connected to the expansion interface. The connections are detailed in the IN and OUT section of the manual as VIDEO and 0 VOLTS respectively. By the time the Issue 3 model was prepared, monitor prices had fallen, and the demand was recognised – so the connection has been made. With a proper lead this output should drive a suitable monitor perfectly well. However, if it's used at the same time as the TV output, the signal to both will be noticeably weakened. It has been suggested that taking a signal from the 0 VOLTS and Y lines can manage this without degrading the signal, but we haven't actually seen it done.

Monitor output the expensive way

Anything that can reasonably be called a monitor relies on this process of taking signals directly to the picture tube electronics. But they aren't all the same. Some rely on this alone, including most of the 'TV monitors' – which are ordinary TV sets fitted with extra circuitry and inputs to take composite or RGB signals. Many were really designed for video work – they won't give you better resolution than your TV, but they will give you sound facilities and a steadier, smoother picture. Truly 'dedicated' monitors don't usually have sound facilities, but they have a whole range of extra facilities to get more detail out of the signal. The snag is that they can't be used with videos or as TV tuners, which makes them an expensive luxury. Even the cheapest monochrome monitors cost more than a small black and white TV – more if they have green or amber screens, which are kinder to the eyes. At least you don't need a licence for them! These monitors suit pure text pretty well, but *Valhalla* or *Manic Miner* are going to look rather odd in shades of glowing green or amber!

Fortunately, they don't have to be. For £285 – which is admittedly a lot of anyone's money – you can buy the **Microvitec Cub** monitor with Spectrum interface. We used a Cub monitor to write this book and we can thoroughly recommend it. It's almost painful to go back to a TV! On the Cub, the Spectrum picture is steady, clear, and pin-sharp – you can just about count the individual pixels! The only problem we experienced was heavy static build-up that sometimes crashed the program in the computer. The Cub gets its signals, logically enough, from the user port, and its interface there has a full through bus that proved to be no hindrance whatsoever to connecting other peripherals. The only put-off is the price, and the minor nuisance that it can't handle the BRIGHT command. Like many professional monitors, it has no sound facility, so you can't use it with units that need TV sound such as the Currah.

Monitor output the patient way

Just as this book was going to press, **Miracle Systems** announced a new RGB interface for the Spectrum. Like the Cub's, it connects direct to the user port. Unlike the Cub's, it doesn't use any of the video signals generated by the computer. Instead, it carries its own 8K RAM, which holds a copy of the Spectrum's video display file and uses this as its picture source. The interface is expected to drive both TTL and linear RGB monitors, and would even let you have a TV picture at the same time without any noticeable loss of signal strength. On its original release this interface cost £74.75. Now it's expected to cost about £50. The unit will also include a loudspeaker and volume control to amplify BEEPs, and a MIC socket that will allow you, as the Fuller Box does, to keep the MIC lead permanently connected. Used with a linear monitor you'll even be able to get that BRIGHT command!

However, if you are patient a little longer than that – and willing to take a gamble – you might do even better. TV manufacturers seem (at long last!) to be recognising a trend in the computer market, and it may soon be possible to get TV/monitors that are almost as good as the Cub – and a lot cheaper.

And finally ...

... a quiet word of warning, not from me, but from a friend in the medical electronics field. Every cathode-ray tube, in a TV, a monitor, or anything else, generates X-rays. On a monochrome tube it's insignificant, but a colour signal is produced by three guns (or one very powerful one) and generates just a little bit more. Normally this wouldn't hurt a flea, and anyway you don't spend all day every day hunched in front of a TV screen. But some people – like computer book authors – do, and over a period of time even the weakest X-ray dosage can become a health hazard.

Chapter Eleven
Looking Ahead

Where do I go from here?

If this book now seems to have exhausted all the possibilities for adding new devices and new functions to the basic Spectrum, then it's only fair to warn you that everything you've read about so far constitutes the tip of an extremely large and rapidly expanding iceberg. The breadth and range of peripherals for the Spectrum is growing too fast for any one writer – or even any three writers – to keep up with. Not all the enquiries I sent received answers. Some companies had obviously folded. Some had moved. But some, as I discovered at one of the regular ZX Microfairs, were sleeping rather than dead. And one of the most fascinating products to be revived from the grave is certainly the Basicare system mentioned very briefly in Chapter 4. Then I said that it offered memory expansion up to four kilobytes (4K), but only for advanced users. The fact is that Basicare's system is far more than a memory expansion device. It's a toolkit for tailoring your Spectrum to do almost any job that can be handled by a microprocessor. And that means *control*.

The Spectrum in charge

The subject of 'computers in control' is very large – large enough for a TV series – so I can't hope to cover it in one short chapter. On the other hand, most of the necessary theory was covered in Chapter 1; and we've already seen, especially in Chapter 6, how the Spectrum's OUT command can be used to control other microchip circuits and produce, in that particular case, controlled sounds. Looking at joystick interfaces, we've studied the IN command, and seen how the Spectrum can read in codes generated by other equipment and use them in its programs. In Chapter 7 we looked at the RD Digital

Tracer and saw how it acted as an analogue to digital converter, translating voltages generated by potentiometers into numbers between 0 and 255. It's only a short step from these ideas to a fully-fledged A/D converter that converts signals, or voltage fluctuations, or movements, or any other external source of information, into numbers – and uses numbers generated by the Spectrum to throw switches, move motors, or a thousand and one other useful possibilities. This isn't just pie in the sky for the average Spectrum owner; once you've grasped the basic principles, it's simple. And the hardware you need for it already exists; I've used it.

Making it happen

Let's start with a simple 'for instance'. Suppose you were going out and you wanted the lights to come on in your sitting-room when it got dark. You'd check the time of sunset in your diary and find, say, that it was 7.30 p.m. Then you might use a special time switch, set to 19.30, which would switch on at the appropriate moment. You could actually do that job quite simply on the Spectrum.

Start by connecting an analogue-to-digital converter and a switching unit to the user port. I used the ADC 8.2 and REL 4.2 units from Harley Systems. Next, connect a light-sensitive cell to the converter. Using the instructions supplied with your converter you can quite easily get a screen display showing the readings produced by the cell. Find out the readings given in daylight and when the room is dark (draw the curtains!). Now connect up the light to the switch unit – this is quite simple, but do make sure that the unit is designed for mains voltages! Now all you need do is write a simple loop program that checks the reading coming in from the light meter using an IN command. If the reading drops below a certain level, then the program moves on to an OUT command that operates the switch and turns on the light. And there you are. You can keep the program ready for use anytime, and forget about looking up sunset times in your diary. It's practical, simple, and effective – after all, you don't use your computer when you're out! Incidentally, Fig. 11.1 shows a graph display in response to input from a light-sensitive cell.

But suppose you go one stage further? There is space on the REL 4.2 to connect four different mains devices, and space on the ADC 8.2 for eight 'sensors' – anything from heat measuring devices to movement detectors. With one fairly simple program you could use

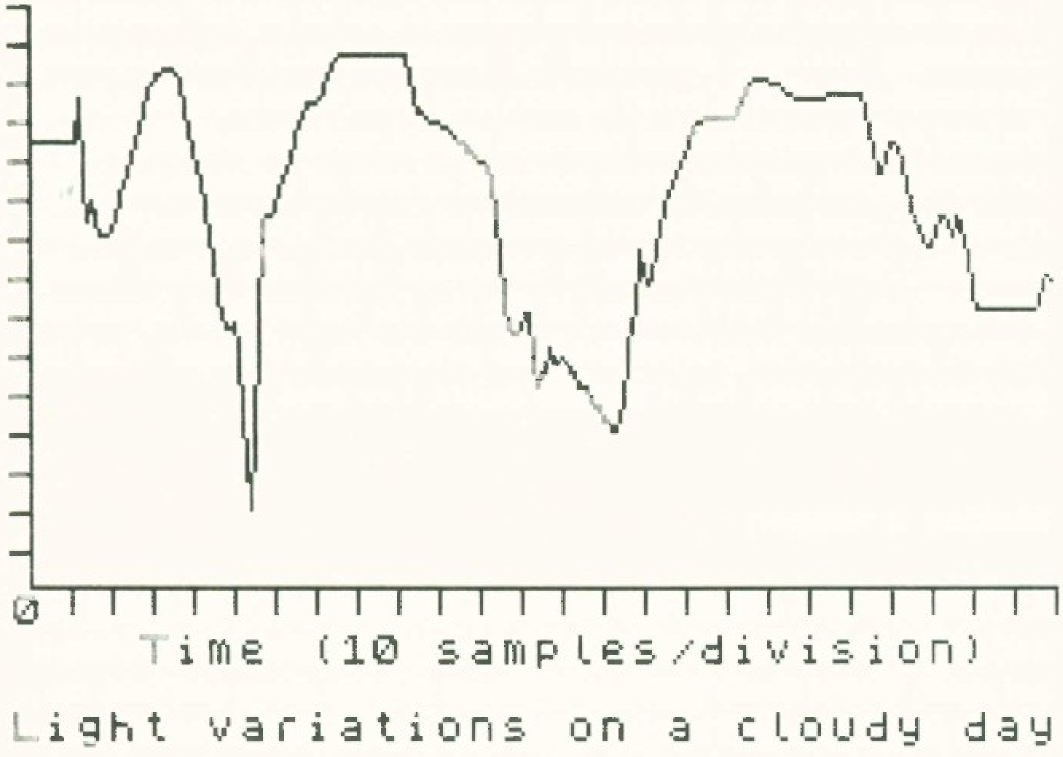

Fig. 11.1. A graph display in response to input from a light-sensitive cell connected to the Blackboard Electronics A/D Converter.

the Spectrum not only to control the lights, but also to set off an alarm if it detects movement in the house, or if a heat sensor detects a fire.

In much the same way, your Spectrum could control a robot. Most robots in production today (and there are plenty) use something called a *stepper motor*; its movements can be precisely controlled, one 'step' at a time, by applying controlled bursts of power. For instance, if your analogue-to-digital converter was connected to output port 131 you could arrange for OUT 131,90 to turn the motor through 90 degrees – and your robot would duly turn 90 degrees in response. Minus numbers could be used to reverse direction: so OUT 131, −90, would turn him back to where he was before. This isn't fantasy: stepper motors can be bought, and you can always steal some of the kids' Fischer Technik to build the robot! If the kids want to play too, how about writing a program to control a model railway? A switch unit like the REL 4.2 could easily perform simple jobs like switching points and signals.

But the available hardware doesn't stop there. I also tested another analogue-to-digital converter from Blackboard Electronics (distributed by Philip Harris) which included a microphone and a full ECG kit (as the deadline for this book drew ever nearer I was able to monitor my incipient heart attack). The demo tape included a 'fun' program that involved shouting into the microphone in order to keep a helicopter on screen 'airborne' – but it wouldn't take an ace programmer to link, for instance, the microphone to a piece of software that gave an insistent flashing warning whenever the kids' stereo was making too much noise, or even (using a switch unit) turned it off! (Cruel but effective!) It's also worth remembering that the William Stuart Big Ears unit is really an analogue to digital converter – the software simply sets up patterns of numbers corresponding to the sounds of a word in memory, and when the unit gets a reasonable match for that word, it 'recognises' it.

Glanmire Electronics supply a time controller that can simply be plugged into the user port. You can use it two ways. By taking readings from it and checking them in the program, your Spectrum can become a time switch. By running a different program and incorporating readings from the time controller, you can get a precise record of when something happened.

All these ideas may sound mildly useful on their own – but imagine combining them! That's possible, too, especially if you use EPROMs. EPROM stands for Erasable Programmable Read Only Memory, and it's a very special kind of microchip that can be programmed with your own software routines and then switched in to replace part or all of the Spectrum ROM – rather like the cartridge ROMs for Interface 2. The difference is that you reach it with an OUT command. You can have a whole nest of EPROMs connected to the Spectrum and access them one at a time – a whole program library waiting to be cut in. My sample board from Orme Electronics included a renumbering routine, another that could delete a selected block of program lines (very handy) and another that would scroll graphics from side to side. EPROM programming is a job for the knowledgeable, and certainly not something that I would tackle tomorrow – but it's possible, and it opens up a whole new area of control for the Spectrum. The real beauty of an EPROM is that it doesn't use any of the Spectrum's own memory space, so you can have a fairly complicated program in memory and call up elaborate subroutines on EPROM that you'd never be able to fit into the computer otherwise.

Building the tower of power

I started this chapter with the Basicare system, and now that we've seen what's possible in this area it's time to look at it again. The foundation of the Basicare system is the so-called Persona unit – and to fit it to a 48K Spectrum you start by effectively cutting out the 32K of memory you bought the thing for! On its own the Persona does very little, so you may wonder why you should perform this act of sacrilege, but in fact the unit is a buffered (i.e. protected) interface for all the other modules in the Basicare range. Once it's fitted, you can start stacking . . . and stacking . . . and stacking. (That theoretical 4000K would form quite a formidable tower.) Mind you, after five units you're going to need some more power from somewhere – the poor little Spectrum mains unit can only manage just so much!

So what, you may ask, are the units? The short answer is almost everything I've discussed above, including a 64K RAMpack, an 8-channel analogue to digital converter, a sound synthesiser, a clock, an 8K ROM unit that can replace part of the Spectrum ROM when required to do so, a 24-line input/output board, a Centronics interface, a Minimap module that lets you select which chunk of all this you need to work with at any given time, and so on and so on. If you seriously want to turn your computer into the heart of a control and communications system, then this is one very well-structured way to do it. Believe it or not, the system was originally designed for the ZX81 – on a big Basicare system it must have looked like a swallow's nest on the side of a skyscraper. The Basicare system is not a toy, and it's not for dabblers – you need to know what you are doing. But it's odd how computers bought as toys by dabblers become serious tools for serious users.

The Spectrum as chameleon

Throughout this book we have seen the Spectrum, once almost despised as a toy, turned to a surprising and remarkable range of practical, not-so-practical, and downright impractical purposes. My Spectrum, in the course of writing the book, has been a word processor, a data processor (it was the only way to keep track of all those manufacturers!), a graphics generator, a database access keyboard (to Prestel), an electronic mailbox, a games machine, a domestic controller, a sweet talker and music maker, a recordkeeper, and a constant source of surprise and enjoyment. It has been a test-bed for new ideas and new equipment, a source of relaxation and a superbly flexible creative tool. I've learned new techniques and developed new ideas. Some of them – most of them – I have tried to pass on in this book.

The Spectrum isn't just a small black box full of circuitry. It's an endless source of interest, new ideas, and new forms of enjoyment. If some of the things you've been reading about have caught your attention, go out and try them for yourself. And if all this detailed analysis of hardware, and the ins and outs of programming for it, has left you feeling a bit weary, why not try out the Spectrum Adventure in Appendix 1?

Good luck – and good computing!

Appendix 1

The Spectrum Adventure

You are trapped in a gigantic powerhouse, sizzling with energy. And all you can see is a lever on the west wall. If and when you escape the powerhouse, you'll find a strange and seemingly random world of embassies, telephone exchanges, corridors, sound and vision studios, and bus stations. And if you've been reading this book, some of them at least should start to sound very familiar.

The Spectrum Adventure isn't an adventure game in the conventional sense. It's an attempt to show, in a reasonably entertaining way, just what can be done with the Spectrum and with Spectrum peripherals. Because it can deal with five different peripherals – a joystick, a light pen, a speech synthesiser, twin Microdrives, and a networked Spectrum – it's also fairly complex. If you're interested, please don't try to enter it all in one day – you will finish up frustrated, tired and angry, and your program will almost certainly be severely infested with bugs. To help you understand what's going on the listing is liberally scattered with REM statements, and important variable names are called things like 'message' and 'noun' rather than being cloaked behind mysterious 'm's and 'n's. The variables j and k are normally used in FOR...NEXT loops (there are plenty in this program). As far as possible all INPUT statements are error-trapped. This makes the programming harder, but it also makes it harder for inexperienced players to crash the game during play.

If you have any problems with the Adventure please don't phone Granada – they're publishers, not computer programmers! Write to me, care of Granada Technical Books, enclosing a stamped addressed envelope, and I will do my best to sort out your difficulties. But before you write, check very carefully that you have entered the program correctly – it's boring to do this, but if you have made a mistake it's unlikely that I'll be able to help you. If you're really stuck, a full version of the program including *all* the arrays, is available on cassette. Again, write to me care of Granada Technical Books.

Vocabulary

To keep the listing down to a length that would fit in this book, I have used only 20 words for the command vocabulary of this adventure. In order, they are:

N	S	E	W	NE	SE	SW	NW
GET	DROP	LOOK	MOVE	OPEN	READ	INVENTORY	BOOK
PROGRAM	KEY	FILE	BUS				

Notice that entering directions in full (e.g. NORTH) will not work. Directions are entered as a single entry – so to go NW you enter "NW", not "MOVE NW". Commands must be entered in the form VERB + NOUN, e.g. "GET BOOK" is a valid command, "GET THE BOOK" won't work, and "GET BOOK AND FILE" will only get you the book. The commands LOOK, MOVE, and INVENTORY don't need nouns with them: LOOK will (perhaps) tell you more about a room, MOVE will bring up a screen display showing you the exits open to you, and INVENTORY (or INV) will tell you what you're carrying.

Rooms

There are 17 rooms in the Spectrum Adventure, as follows:

1 The Power House.
2 A control room.
3 The Spectran Embassy.
4 A telephone exchange.
5 A bus station. In one corner is a Spectrum.
6 A high gallery.
7 A sound studio.
8 A fast-moving conveyor belt.
9 An empty room. A huge clock fills one wall.
10 A TV studio.
11 A TV transmitter.
12 You are inside a glass wall.
13 A flat black plain with low grey blocks visible.
14 An endless row of filing cabinets.
15 An unbearably hot room.
16 A sound stage.
17 A corridor with a locked door to the N.

You can add to these descriptions and embroider them as you wish, but they are *not* included in the listed program. You must either enter them yourself using READ...DATA statements as in subroutine 7000, or work on the Microdrive program listed earlier in the book so that you can create the necessary DATA array, r$, and store it on Microdrive. The game listing assumes you have Interface 1 and Microdrive.

If you don't think you can tackle these little problems, then I don't recommend you to try the game – you could find yourself in some difficulty. However, for those of you still with me, let me explain how it works.

Anatomy of an adventure

The Spectrum Adventure depends on a series of arrays that store information in a carefully ordered sequence. These arrays are as follows (dimensions in brackets):

Room array	r$(17,96)	"rooms"
Movement array	m(17,8)	"moves"
Vocabulary array	w$(20,2)	"words"
Speech array	t(20,10)	"talk"
Message array	m$(2,32)	"messages"

This means that movement array no. 6 is for the room described by room array no. 6, and speech array no. 5 is the programming to 'say' the word in vocabulary array no. 5. As you have seen, the first eight words in the vocabulary are directions, so you won't be surprised to find that the eight elements of each movement array correspond to those directions. For instance, m(16,1) gives the room number you will reach if you go NORTH from room 16.

The program is carefully structured, as follows:

10–50	Load data from subroutine 7000 and from Microdrive. Set up all variables.
100–200	Check up on peripherals being used. If light pen is used, load machine code from Microdrive (so make sure you put the necessary machine code on the game cartridge!). If network is in use, GO SUB 4000.
1000–1010	If second Microdrive is in use, load screen image (locSCREEN$). Print room description, which is r$(loc).
2000–2910	The core of the program. This section analyses the

command string (c$) that you enter from the keyboard by taking the first two letters and checking them against the two-letter strings stored in w$. If it finds a match, then 'verb' takes its value from the array number of w$. Thus, if you enter "GET BOOK" then the first two letters are "GE", which matches w$(9), so the value of 'verb' will be 9. After finding the first space in c$, the program then takes the next two letters and checks them against the nouns in w$ – if it finds one, then 'noun' becomes the appropriate array number (so in this example "BO" will equal w$(16), and 'noun' will have the value 16). Once these values have been found, the program goes to line 2000+(10*verb), in this case line 2090, and checks (a) whether the command is valid and (b) what to do if it is. If it isn't, message 1 ('Sorry, but you can't do that') is returned. If message = 0, the computer prints 'OK' and carries out the command. If speech synthesis is being used, the speech subroutine is called at 2065.

3000–3180 This subroutine checks on some of the variables created in the main program, according to which room you happen to be in at the time (i.e. if you're in room 16 (loc = 16) then the program goes to 3160). If you're carrying the wrong object in the wrong place, or you've made a mistake earlier on, this routine doles out the punishment!

3500 Success! This routine is for game winners only.

3600 Failure! This takes care of the losers!

4000 The network subroutine turns the game into a sort of ping-pong match. Each player must wait for his opponent to move, but he is told each move his opponent makes. Objects picked up by one player are not available to the other player until they're dropped – and they will be in the room where the first player dropped them.

5000 This routine sets up a screen menu showing possible exits (if any) from any given location. It will work with a light pen OR a joystick OR from the keyboard, but you must choose which to use at 1000. The programming here is fairly intricate. However, if you can unravel it you'll find it useful in your own light pen/joystick menu routines.

7000 This subroutine fills all the data arrays dealt with from inside the program (i.e. everything except the "rooms" array, which you must set up yourself).

8000 Speech subroutine. This checks the speech array for the codes to match the command words sent from 2065, and sends them OUT to the speech unit. Remember to change the OUT value in this routine to suit your own unit, or you'll get nothing at all!

9000 These simple subroutines take a reading for Lno (Light pen no.) which are used in the direction routines at 5000.

9300 This subroutine does the same for the joystick readings if you're using the joystick option, but it will only work with joysticks that can duplicate the cursor keys. If yours doesn't, you will need to reprogram this section.

Using this simple skeleton, you should be able to work the rest out for yourself – but not *too* quickly, I hope! After all, if you can see just how the program works you can cheat – and that would spoil *my* fun. One word of advice. When you catch a bus, do think about where it might be going ...

The Spectrum Adventure program listing

```
   7 REM ************************
   8 REM LOAD PREPARED DATA
   9 REM ************************
  10 CLEAR 63029: POKE 23658,8:
REM Set CAPS LOCK
  20 BORDER 1: PAPER 1: INK 7: C
LS
  30 GO SUB 7000: REM Fill data
arrays
  40 LOAD *"m";1;"rooms" DATA r$
()
  50 LET player=0: LET inv=0: LE
T move=0: LET verb=0: LET noun=
0: LET lever=0: LET knob=0: LET
computer=0: LET key=0: LET bus=1
: LET file=0: LET book=0: LET bo
ttle=1: LET end=0
  97 REM ************************
  98 REM SELECT PERIPHERALS
  99 REM ************************
```

```
 100 PRINT "SPECTRUM ADVENTURE"'
'"First, a check on the add-ons
"'"you're using.  Please enter Y
 orN for each."
 110 RESTORE 130: DIM p(5): FOR
j=1 TO 5: READ p$: PRINT AT 12,
0;p$: INPUT "(Y/N)",a$: IF a$="Y
" THEN  LET p(j)=1
 120 NEXT j
 130 DATA "Joystick","Light pen"
,"Speech synthesiser","2nd Spec
trum for networking","2nd Microd
rive for your own     graphics"
 140 LET joy=p(1)
 150 LET pen=p(2): IF pen  THEN
 LOAD *"m";1;"menu"CODE : REM T
his will only work if you have r
ecorded the light pen machine co
de on the Adventure Microdrive c
artridge
 160 LET speech=p(3)
 170 LET net=p(4)
 180 LET drive=p(5)
 190 IF net THEN  CLS : PRINT "M
AKE SURE YOUR OPPONENT IS ALSO
RUNNING THE GAME": GO SUB 4000
 200 LET loc=1: GO SUB 1000: GO
TO 2000
 997 REM ************************
 998 REM DESCRIBE LOCATION
 999 REM ************************
1000 CLS : IF drive THEN  LOAD *
"m";2;STR$ locSCREEN$
1010 PRINT AT 8,0;r$(loc)
1997 REM ************************
1998 REM COMMAND SUBROUTINE
1999 REM ************************
2000 LET verb=0: LET noun=0: DIM
 n$(2): DIM c$(12): INPUT "What
 do you want to do?"'c$: PRINT c
$
2010 FOR j=1 TO 15: IF c$( TO 2)
=w$(j) THEN  LET verb=j: LET j=
15
```

```
2020 NEXT j: IF verb<9 AND verb>
0 THEN  LET n$=w$(verb): GO TO
5050
2030 FOR j=3 TO 12: IF c$(j)=" "
 THEN  LET n$=c$(j+1 TO j+2): L
ET j=12
2040 NEXT j
2050 FOR j=16 TO 20: IF n$=w$(j)
 THEN  LET noun=j: LET j=20
2060 NEXT j
2065 IF speech  AND verb THEN  L
ET j=verb: GO SUB 8020: IF noun
 THEN  LET j=noun: GO SUB 8020
2070 LET message=0: IF verb  THE
N  GO SUB 2000+verb*10
2074 IF message=0 AND verb THEN
 GO TO 2080
2076 GO SUB 2900: GO SUB 3000: G
O TO 2000
2080 PRINT ''"OK"''
2083 IF net AND player=1 THEN  G
O SUB 4100: GO SUB 4200
2084 IF net AND player=2 THEN  G
O SUB 4200: GO SUB 4100
2085 GO SUB 3000: GO TO 2000
2089 REM ********GET***********
2090 IF noun=0 THEN  LET message
=1: RETURN
2091 IF noun=20 AND loc=5 THEN
RETURN
2092 IF o(noun)<>loc THEN  LET m
essage=1: RETURN
2094 IF o(noun)=loc AND inv<2 TH
EN  LET o(noun)=0: LET inv=inv+
1: RETURN
2096 LET message=2: RETURN
2099 REM ********DROP**********
2100 IF noun=0 THEN  LET message
=1: RETURN
2102 IF o(noun)=0 THEN  LET o(no
un)=loc: LET inv=inv-1: RETURN
2104 LET message=1: RETURN
2109 REM ********LOOK**********
2110 CLS : PRINT r$(loc)
```

```
2112 FOR j=16 TO 19: IF o(j)=loc
 THEN  PRINT ''"You can also se
e:": LET j=19: NEXT j: GO TO 211
6
2114 NEXT j: RETURN
2116 RESTORE 2154: FOR k=1 TO 4:
 READ object: READ o$: IF o(obj
ect)=loc THEN  PRINT o$
2118 NEXT k: RETURN
2119 REM ********MOVE**********
2120 IF loc=5 AND computer THEN
 LET verb=9: LET noun=20: RETUR
N
2122 IF loc=5 THEN  LET message=
1: RETURN
2124 GO TO 5000
2129 REM ********OPEN**********
2130 IF (noun=16  AND book) OR (
noun=19 AND file) THEN  PRINT '
'"It's already open!": RETURN
2132 IF noun=16 THEN  LET book=1
: RETURN
2134 IF noun=19 THEN  LET file=1
: RETURN
2136 LET message=1: RETURN
2139 REM ********READ**********
2140 IF noun=16  AND book THEN
PRINT "It's a book about the Sp
ectrum, but the title and conten
ts pagesare missing.": RETURN
2142 IF noun=17 THEN  PRINT "How
 good are you at reading     Mi
crodrive cartridges?": RETURN
2144 IF noun=19 AND file THEN  P
RINT "It says 'This is a blank
file'.": RETURN
2146 LET message=1: RETURN
2149 REM ********INVENTORY*****
2150 PRINT "You have:": LET k=0
2152 RESTORE 2154: FOR j=1 TO 4:
 READ object: READ o$: IF o(obj
ect)=0  THEN  PRINT o$': LET k=1

2154 DATA 16,"A paperback book",
18,"A golden key",19,"A brown m
```

```
anila file",17,"A computer progr
am"
2156 NEXT j: IF k=0 THEN  PRINT
"nothing at all"
2158 RETURN
2900 IF verb=0 THEN  LET message
=1
2910 PRINT ''m$(message)'': RETU
RN
2997 REM ************************
2998 REM STATUS SUBROUTINES
2999 REM ************************
3000 GO SUB 3000+loc*10: RETURN
3009 REM ********Room 1*********
3010 LET a$=""
3012 IF verb=11 THEN  PRINT ''"T
here is a lever on the west
wall.  Do you want to pull it?
(Y/N) ": INPUT a$
3014 IF a$="Y" AND lever=0 THEN
 LET lever=1: LET a$=""
3016 IF a$="Y" AND lever=1 THEN
 LET lever=0: RETURN
3017 IF move<>2 AND (lever<>1 OR
 knob<9) THEN  LET move=0
3018 RETURN
3019 REM ********Room 2*********
3020 IF verb=11 THEN  GO SUB 302
4
3022 RETURN
3024 PRINT ''"There's a knob on
the wall with markings from 0 t
o 9. It's set  at ";knob'': INPU
T "Do you want to change it? (Y/
N)";a$: IF a$<>"Y" THEN  RETURN
3026 INPUT "What setting? ";a$:
IF CODE a$(1)<48 OR CODE a$(1)>
58 THEN  GO TO 3026
3028 LET knob=VAL a$: RETURN
3029 REM ********Room 3*********
3030 RETURN
3039 REM ********Room 4*********
3040 RETURN
3049 REM ********Room 5*********
3050 IF computer=0 THEN  LET bus
```

```
=INT (RND*17)+1: PRINT ''"A num
ber ";bus;" bus arrives."'"Do yo
u want to get on? (Y/N)": INPUT
a$: IF a$="Y" THEN  PAUSE 200: P
RINT "The bus stops here. You ge
t off.": PAUSE 150: LET loc=bus:
 RETURN
3052 IF computer AND verb=9 AND
noun=20  THEN  PRINT #0: PAUS
E 150: CLS : PRINT "Which bus do
 you want to catch?": GO SUB 305
6: RETURN
3054 IF verb=10 AND noun=17 THEN
  LET computer=1: CLS : PRINT "
The screen lights up. It reads:"
: PRINT ''"BUS TERMINALS AS FOLL
OWS:": FOR j=1 TO 17: PRINT j'r$
(j): NEXT j: RETURN
3055 RETURN
3056 INPUT a$: IF CODE a$(1)<49
OR CODE a$(1)>57 THEN  GO TO 30
56
3057 IF VAL a$<=17 THEN  LET bus
=VAL a$: CLS : PRINT "A number
";bus;" bus arrives and"'"takes
you to your chosen        destin
ation.": LET loc=bus: PAUSE 150:
 CLS : LET message=0: RETURN
3058 GO TO 3056
3059 REM *********Room 6********
3060 RETURN
3069 REM *********Room 7********
3070 RETURN
3079 REM *********Room 8********
3080 LET end=end+1: IF end<3 THE
N  RETURN
3082 PRINT "You are being crushe
d to death  by the drive mechan
ism of your  cassette recorder.
 Nasty,       isn't it?": PAUSE 2
00: GO TO 3600
3089 REM *********Room 9********
3090 RETURN
3099 REM *********Room 10*******
```

```
3100 RETURN
3109 REM *********Room 11*******
3110 IF bottle=0 THEN  LET end=e
nd+1
3112 IF end<3 THEN  RETURN
3114 LET end=0: PRINT ''"The win
e has pixillated you. Youare be
ing transformed into      binary
 digits...": PAUSE 150: LET loc=
12: RETURN
3119 REM *********Room 12*******
3120 LET end=end+1: IF end<3 THE
N  RETURN
3122 PRINT "Unfortunately the pr
ogrammer hasforgotten you.  He
enters CLS...": PAUSE 150: GO TO
 3600
3129 REM *********Room 13*******
3130 RETURN
3139 REM *********Room 14*******
3140  IF o(18)=0 AND verb=11 THE
N  GO TO 3143
3142 RETURN
3143 PRINT ''"There are three dr
awers here.   Which one will yo
u open? (Key 1,2, or 3 - any oth
er key to enteranother command)"
: INPUT a$: IF CODE a$<49 OR COD
E a$>51 THEN  RETURN
3144 LET j=INT (RND*3)+1: IF j=V
AL a$ AND o(16)=255 THEN  LET o
(16)=loc: PRINT ''"You can see a
 paperback book": RETURN
3145 IF j=VAL a$ THEN  PRINT ''"
The drawer is empty.": RETURN
3146 PRINT ''"A bug leaps out an
d bites you.  You lose everythi
ng you are     carrying.": LET i
=0: FOR j=16 TO 19: IF o(j)=0 TH
EN  LET o(j)=INT (RND*10)+1
3147 NEXT j: RETURN
3148 PRINT "A bug leaps out and
bites you.  You lose everything
 you are     carrying.": DIM i(2
```

```
): FOR j=16 TO 19: IF o(j)<>255
THEN  LET o(j)=INT (RND*10)+1: R
ETURN
3149 REM *********Room 15*******
3150 LET end=end+1: IF end<3 THE
N  RETURN
3152 CLS : PRINT "Ouch! You've l
anded on the heat sink where yo
u are slowly fryingto death...":
 PAUSE 150: GO TO 3600
3159 REM *********Room 16*******
3160 IF verb=11 AND bottle THEN
 PRINT ''"There's a bottle of w
ine here.  Fancy a drink? (Y/N)"
: INPUT a$: IF a$="Y" THEN  LET
bottle=0: PRINT ''"Aah... that's
 better!": RETURN
3162 RETURN
3169 REM *********Room 16*******
3170 INPUT "Do you want to open
the door?   (Y/N) ";a$: IF a$="
Y" THEN  GO TO 3174
3172 RETURN
3174 IF o(16)=0 AND o(18)=0 AND
computer=1 THEN  GO TO 3178
3176 PRINT #0;"You can't -- yet!
": PAUSE 150: RETURN
3178 PRINT "Are you sure? (Y/N)"
: INPUT a$: IF a$="Y" THEN  GO
TO 3500
3497 REM ***********************
3498 REM SUCCESS!
3499 REM ***********************
3500 CLS : PRINT  FLASH 1;"CONGR
ATULATIONS: YOU'VE SOLVED  THE
SPECTRUM ADVENTURE"'' FLASH 0;"N
ow that you have the chapter   h
eadings and index for 'The     S
pectrum Add-On Guide', read on!"
3510 IF net THEN  OPEN #4;"n";3-
player: PRINT #4;18;"Your oppon
ent has solved the    adventure.
": CLOSE #4
3520 GO TO 3520
```

```
3597 REM ***********************
3598 REM FAILURE!
3599 REM ***********************
3600 CLS : PRINT  FLASH 1;"YOU A
RE DEAD"; FLASH 0; IF net=0 THE
N  PRINT ''"Want to try again? (
Y/N)": INPUT a$: IF a$="Y" THEN
 CLS : GO TO 100
3610 IF net THEN  OPEN #4;"n";3-
player: PRINT #4;0: PRINT #4;"Y
our opponent has failed: you'reo
n your own now!": CLOSE #4
3620 GO TO 3620
3997 REM ***********************
3998 REM NETWORK SUBROUTINES
3999 REM ***********************
4000 OPEN #4;"n";0: PRINT #4;"1"
: CLOSE #4: OPEN #4;"n";0: INPU
T #4;p$: CLOSE #4
4010 IF p$="1" THEN  OPEN #4;"n"
;0: PAUSE 5: PRINT #4;"2": LET
player=1
4020 IF p$="2" THEN  LET player=
2
4030 FORMAT "n";player: CLOSE #4
: RETURN
4100 OPEN #4;"n";3-player: PRINT
 #4;loc: PRINT #4;c$
4110 IF loc=14 THEN  FOR j=16 TO
 19: PRINT #4;o(j): NEXT j
4120 CLOSE #4: RETURN
4200 OPEN #4;"n";3-player: INPUT
 #4;loc2: INPUT #4;d$: PRINT ''
"Your opponent has entered:"'d$:
 CLOSE #4
4210 IF d$( TO 2)="GE" THEN  GO
SUB 4300
4220 IF d$( TO 2)="DR" THEN  GO
SUB 4400
4230 IF loc2=18 THEN  CLS : PRIN
T d$''"Better luck next time!":
 STOP
4240 IF loc2=0 THEN  CLS : PRINT
 d$''"You're on your own now -
```

```
and thebest of British luck!": P
AUSE 200: LET net=0: GO TO 1000
4250 IF loc2=14 THEN  FOR j=16 T
O 19: INPUT #4;o2: IF o(j)<>0 A
ND o2<>0 THEN  LET o(j)=o2
4260 NEXT j
4270 CLOSE #4: RETURN
4300 FOR j=16 TO 20
4310 IF w$(j)=d$(5 TO 6) AND o(j
)<>0 THEN  LET o(j)=255: LET j=
20
4320 IF w$(j)=d$(5 TO 6) AND o(j
)=0 AND player=2 THEN  PRINT "H
ard luck, he was too fast for  y
ou after all.  He's got it now."
: LET o(j)=255: LET j=20
4330 NEXT j: RETURN
4400 IF d$( TO 2)="DR" THEN  FOR
 j=16 TO 20
4410 IF d$(6 TO 7)=w$(j) THEN  L
ET o(j)=loc2: LET j=20
4420 NEXT j: RETURN
4997 REM ************************
4998 REM MOVEMENT MENU
4999 REM ************************
5000 CLS : DIM d(16): LET k=0: L
ET line=4: PRINT "Exits as foll
ows:"'': FOR j=1 TO 8
5010 IF m(loc,j) THEN  PRINT AT
line,2;"A ";w$(j)'': LET d(line
/2)=m(loc,j): LET line=line+2: L
ET k=k+1
5015 NEXT j: IF k=0 THEN  PRINT
AT line,2;"There are no visible
 exits.": LET line=line+2
5020 PRINT AT line,2;"A For any
other command": LET d(line/2)=2
000
5025 IF joy THEN  GO SUB 9300: G
O SUB 3000: GO TO 5070
5030 IF pen THEN  GO SUB 9200: G
O SUB 3000: GO TO 5070
5040 DIM n$(2): INPUT "Enter you
r choice as shown (key ENTER fo
r other commands)",n$
```

```
5050 FOR j=1 TO 8: IF n$=w$(j) T
HEN  LET move=m(loc,j): LET j=8
: NEXT j: GO SUB 3000: GO TO 507
0
5060 NEXT j: GO TO 2000
5070 IF move=0 THEN  PRINT #0;"S
orry, you can't go that way.":
PAUSE 150: GO TO 5000
5080 IF speech  THEN  GO SUB 800
0
5085 LET loc=move
5090 IF net AND player=1 THEN  G
O SUB 4100: GO SUB 4200: PAUSE
150
5095 IF net AND player=2 THEN  G
O SUB 4200: GO SUB 4100: PAUSE
150
5100 GO SUB 1000: GO TO 2000
6990 REM ***********************
6991 REM DATA FILES These can be
6992 REM set up on Microdrive to
6993 REM conserve memory space
6994 REM ***********************
6995 REM
6997 REM ***********************
6998 REM VOCABULARY ARRAY
6999 REM ***********************
7000 DIM w$(20,2)
7010 RESTORE 7020: FOR j=1 TO 20
: READ w$(j): NEXT j
7020 DATA "N ","S ","E ","W ","N
E","SE","SW","NW","GE","DR","LO
","MO","OP","RE","IN","BO","PR",
"KE","FI","BU"
7027 REM ***********************
7028 REM MOVEMENT TABLE
7029 REM ***********************
7030 DIM m(17,8)
7040 RESTORE 7050: FOR j=1 TO 17
: FOR k=1 TO 8: READ m(j,k): NE
XT k: NEXT j
7050 DATA 0,15,0,17,0,2,3,0: REM
  Room 1
7060 DATA 0,0,0,0,0,0,0,1: REM
```

```
Room 2
7070 DATA 0,0,0,4,0,0,14,0: REM
Room 3
7080 DATA 0,0,3,5,0,0,0,0: REM
Room 4
7090 DATA 0,0,0,0,0,0,0,0: REM
Room 5
7100 DATA 7,13,5,9,17,16,3,0: RE
M Room 6
7110 DATA 8,6,0,0,0,0,0,0: REM
Room 7
7120 DATA 0,0,0,0,0,0,0,0: REM
Room 8
7130 DATA 0,0,6,10,0,0,0,0: REM
Room 9
7140 DATA 11,0,9,0,0,0,0,0: REM
Room 10
7150 DATA 12,10,0,0,0,0,0,0: REM
 Room 11
7160 DATA 0,0,0,0,0,0,0,0: REM
Room 12
7170 DATA 6,0,0,0,0,0,0,0: REM
Room 13
7180 DATA 5,0,0,0,3,0,0,0: REM
Room 14
7190 DATA 0,0,0,0,0,0,0,0: REM
Room 15
7200 DATA 0,0,0,0,0,0,0,6: REM
Room 16
7210 DATA 0,0,1,0,0,0,6,0: REM
Room 17
7297 REM ************************
7298 REM SPEECH ARRAY
7299 REM ************************
7300 DIM t(20,10): FOR j=1 TO 20
: FOR k=1 TO 10: READ t(j,k)
7305 IF t(j,k)=0 THEN  LET k=10
7307 NEXT k: NEXT j
7310 DATA 56,58,29,0.: REM North
7320 DATA 55,32,29,0: REM South
7330 DATA 19,55,17,0: REM East
7340 DATA 46,7,55,17,0: REM West
7350 DATA 56,58,29,19,55,17,0: R
EM Northeast
7360 DATA 55,32,29,19,55,17,0: R
```

```
EM Southeast
7370 DATA 55,32,29,46,7,55,17,0:
 REM Southwest
7380 DATA 56,58,29,46,7,55,17,0:
 REM Northwest
7390 DATA 36,7,13,0: REM Get
7400 DATA 33,39,23,9,0: REM Drop
7410 DATA 45,30,41,0: REM Look
7420 DATA 16,31,35,0: REM Move
7430 DATA 53,9,7,11,0: REM Open
7440 DATA 14,19,21,0: REM Read
7450 DATA 12,11,35,7,11,13,23,14
,12,0: REM Inventory
7460 DATA 63,30,41,0: REM Book
7470 DATA 9,39,53,61,39,26,16,0:
 REM Program
7480 DATA 42,19,0: REM Key
7490 DATA 40,40,6,45,0: REM File
7500 DATA 63,15,55,0: REM Bus
7597 REM ************************
7598 REM LOCATION OF OBJECTS
7599 REM ************************
7600 DIM o(20): FOR j=16 TO 19:
READ o(j): NEXT j
7610 DATA 255,13,16,6
7697 REM ************************
7698 REM MESSAGES
7699 REM ************************
7700 DIM m$(2,32): FOR j=1 TO 2:
 READ m$(j): NEXT j
7710 DATA "Sorry, you can't do t
hat.","You can't carry any more
 "
7900 RETURN
7997 REM ************************
7998 REM SPEECH SUBROUTINES
7999 REM ************************
8000 FOR j=1 TO 8: IF move=m(loc
,j) THEN  GO TO 8020
8010 NEXT j: RETURN
8020 FOR k=1 TO 10: OUT 31,t(j,k
): PAUSE 1: IF t(j,k)=0 THEN  L
ET k=10
8030 NEXT k: RETURN
8997 REM ************************
```

```
8998 REM LIGHT PEN SUBROUTINES
8999 REM ***********************
9000 LET Lno=0: PAUSE 20
9010 LET Lno=USR 63109
9020 IF Lno=0 THEN  GO TO 9000
9030 RETURN
9200 GO SUB 9000
9210 LET move=d(Lno/2): IF move>
17 THEN  GO TO move
9220 IF move<1 THEN  GO TO 9200
9230 RETURN
9297 REM ***********************
9298 REM JOYSTICK SUBROUTINES
9299 REM ***********************
9300 LET place=4
9305 IF k=0 THEN  PRINT AT 6,2;
FLASH 1;">"; FLASH 0
9310 IF k THEN  PRINT AT place,2
; FLASH 1;">"; FLASH 0
9315 LET choice=IN 61438: PAUSE
5
9320 IF choice=247 AND place>4 T
HEN  PRINT AT place,2;"A": LET
place=place-2
9330 IF choice=239 AND place<lin
e THEN  PRINT AT place,2;"A": L
ET place=place+2
9340 IF choice=254 AND d(place/2
)<=17 THEN  LET move=d(place/2)
: RETURN
9350 IF choice=254 AND d(place/2
)>17 THEN  GO TO d(place/2)
9360 GO TO 9310
9998 STOP
```

Appendix 2

How to Succeed at Mail Order

by Mike Scott Rohan

And now a word about buying your peripherals by mail order. *Don't!*

No, I'm not just trying to be funny. Mail order businesses provide a very useful service, and there are times when you will have no option but mail order if you have decided on a particular unit. But before you cheerily write and mail your cheque, make sure you know how to protect yourself from the don't-cares, the incompetents, and the genuine crooks.

The 'don't-cares' are basically honest traders. You'll get what you ordered – eventually. But you may have ordered on the basis of an advertisement for a product that was still on the drawing-board, or at prototype stage – and because the manufacturer has got his sums wrong, or tried to stop a competitor cutting away the market for his proposed new product, you are left waiting months and months for something that isn't even properly in production. Some famous names have adopted this policy, and although the products may be worth the wait, it does create a dangerous precedent for the smaller fry. They know you'd rather get the good price – and the good product – than go to all the bother of trying to get your money back. So:

- Phone and check that they are still in business and still selling the same product at the same price with supplies still in stock.
- In your ordering letter, set a definite time limit for delivery (if they insist on their coupon, tack it on as well). The limit should be 28 days at the most.
- Check at least once before the end of your deadline. If nothing has appeared, send a registered letter demanding your money back. Often your order will miraculously appear.

The point is that when you make an order in writing you're negotiating a contract, and if the company takes your money without argument they're accepting your terms. So if the goods don't materialise, and your money doesn't either, talk to your local Citizen's Advice Bureau, the Office of Fair Trading, the Advertising Standards Agency (if the firm is a large one) and – most importantly – the magazine that carried the advertisement, as you should be covered by their mail order protection scheme. This is because you may be dealing with one of the other two categories – an incompetent, or a crook.

Home computing has revived the cottage industry, and there are now hundreds of firms operating from garden sheds and back-street terraces. This is all very good for the economy, but many of these honest folk are not business people, and they have a charming but annoying habit of losing your order somewhere in last week's laundry basket. At least they aren't cashing the cheque and making free with the interest, but if in doubt, badger – especially through the magazines that carry their ads. You may be driving them to a nervous breakdown, but then no one forces them to go into business in the first place. Also, there is just a chance that this kind of well-meaning incompetence is the cover for category number three – the out-and-out crook.

The only way to win with the crooks is to make sure that they never get your money in the first place. Find out all you can about a company before you send your money: has it got a proper address and phone number? If not, forget it. How long has it been trading? Have any magazines had complaints about it? The cost of a few calls could save you a lot of money.

Going bust

It happens – and when it does, you are going to be a long way down the creditor's list. So how can you protect your money?

The best way is to use a legal form of words devised by the National Federation of Consumer Groups (12 Mosley Street, Newcastle-upon-Tyne) who will sell you printed stickers saying:

> *This money is sent on condition that you will hold it as a trustee on my behalf, and that it will remain mine until the goods have been sent to me. If you accept this payment you will be deemed to have accepted this condition.*

It won't stop thieves stealing the money, or frauds misusing it, but it may make them think twice. Their worst enemy, after all, is the person who has the sense to stand up and complain. Another and perhaps better way is to order with a credit card such as Access or Barclaycard. These companies will protect their money – and yours – with an iron fist, but you could have a lot of trouble getting the bad order wiped off your statement!

Support

Lack of easy servicing and back-up is another problem with mail order. The company may be good and offer an excellent guarantee, but it's a long way to go if and when something *does* go wrong. And if you're dealing with incompetents, they may not even have thought of support when marketing their product. However, if something you've bought doesn't perform as advertised, or at all, you're entitled to an instant refund. The law says that anything sold must be fit for the purpose it was sold for. And if it does any harm to your system as a result, you're usually entitled to damages.

Remedies

Your best friend in any mail order case ought to be the magazine that carried the advert. They make money from it, as the law has recognised with the protection scheme – but note that it doesn't seem to apply to classified ads, or if the ad invites you to send for a brochure from which you make your order. If the magazine can't help you try some of the organisations mentioned earlier – and don't be afraid to use the law. The Small Claims procedure is designed for ordinary folk like us, and the more we use it, the better. The more we sit back, the more trusting we are, the harder we make it for good, honest traders to succeed.

Appendix 3

Where to Find Them

ADS
8 Bonchurch Street
Portsmouth, Hants PO4 8RY
(telephone: 0705 823825)

AGF
26 Van Gogh Place
Bognor Regis
W Sussex PO22 9BY
(telephone: 0243 823337)

Bi-Pak
PO Box 6
Ware
Herts
(telephone: 0920 3442)

British Micro
Unit Q2, Penfold Works
Imperial Way
Watford WD2 4YY
(telephone: 0923 48222)

Cambridge Microelectronics
(EPROM products)
1 Milton Road
Cambridge CB4 1UY
(telephone: 0223 314814)

Cheetah
24 Ray Street
London EC1
(telephone: 01-240 7989)

Advanced Memory Systems
Green Lane
Appleton
Warrington WA4 5NG
(telephone: 0925 62907)

Basicare Microsystems Ltd
12 Rickett Street
London SW8 1RU

Blackboard Electronics
17 Beechfield Road
Davenport
Stockport
Cheshire SK3 8SF
(telephone: 061 487 2508)

Cambridge Computing
1 Ditton Walk
Cambridge CB5 8QZ
(telephone: 0223 214451)

CCS (joysticks and interfaces)
PO Box 1W9
Leeds LS16 6NT
(telephone: 0532 670625)

Consumer Electronics Ltd
(Suncom joysticks and others)
Failsworth
Manchester M35 0HS
(telephone: 061 682 2339)

Harley Systems Ltd
The Pepperboxes
Great Missenden
Bucks HP15 9PR
(telephone: 024028 630)

Hilderbay
8–10 Parkway
London NW1
(telephone: 01-485 1059)

ITL Kathmill Ltd
The Old Courthouse
New Road
Chatham
Kent ME4 4QJ
(telephone: 0634 815464)

Jiles Electronics
48 Parkway
Chellaston
Derby DE7 1QA
(telephone: 0332 703892)

Kelwood Computer Cases
Downs Row
Moorgate
Rotherham S60 2HD
(telephone: 0709 63242)

Kempston Microelectronics Ltd
Unit 30, Singer Way
Woburn Road Industrial Estate
Bedford MK42 7AF
(telephone: 0234 852997)

Maplin Electronics Supplies
Maplin Complex
Oak Road South
Benfleet
Essex SS7 2BB
(telephone: 0702 552911)

Micro Power (Add-On Sound Unit)
8 Regent Street
Chapel Allerton
Leeds LS7 4PE
(telephone: 0532 683186)

Micro-Pad
14 Brackley
Queen's Road
Weybridge
Surrey KT13 0BJ
(telephone: 0932 42882)

MicroMyte Communications
Polo House
27 Prince Street
Bristol 1
(telephone: 0272 299373)

Microvitec Ltd (Cub monitor)
P.O. Box 188
Futures Way
Bolling Road
Bradford BD4 7TU
(telephone: 0274 390011)

Minor Miracles (modem)
Ipswich
(telephone: 0473 50304)

Miracle Systems
6 Armitage Way
Kings Hedges
Cambridge CB4 2UE

Morex Peripherals Ltd
172B King's Road
Reading
Berks RG1 4EJ
(telephone: 0734 584238)

Ness Micro Systems
100 Drakies Avenue
Inverness IV2 3SD
(telephone: 0463 790368)

O.E. Ltd
North Point
Gilwilly Industrial Estate
Penrith
Cumbria CA11 9BN

Orme Electronics
2 Barripper Road
Camborne
Cornwall TR14 7QN
(telephone: 0209 715034)

Pinnacle Ltd (BEEP amplifier)
Pinnacle House
Oasthouse Wharf
Orpington
Kent
(telephone: 0689 27000)

Prism Business Systems
Prism House
18–29, Mora Street
London EC1V 8BT
(telephone: 01-253 2277)

RD Laboratories
20 Court Road Estate
Cwmbran
Gwent NP44 3AS
(telephone: 06333 74333)

Saga Systems Ltd
Woodham Road
Woking
Surrey GU21 4DL
Enquiries telephone:
04862 23845

Signpoint Ltd
Unit D, 166A Glyn Road
Clapton
London E5 0JE
(telephone: 01-533 0724)

Stack Computer Services Ltd
290–298 Derby Road
Bootle
Merseyside L20 8LN
(telephone: 051 933 5511)

Network Computer Systems
39 Bampton Road
Luton LU4 0DD
(telephone: 0582 508616)

Orion Data (Micro Command)
3 Cavendish Street
Brighton BN2 1RN
(telephone: 0273 672994)

Petron Electronics
1 Courtlands Road
Newton Abbot
Devon TQ12 2JA
(telephone: 0626 62836)

Primordial Peripherals
89 Herne Road
Bushey
Herts WD2 2LP
(telephone: 01-950 9533)

Protek Computing Ltd
1A Young Square
Brucefield Industrial Park
Livingston
W Lothian
(telephone: 0506 415353)

Ricoll Electronics Ltd
48 Southport Road
Ormskirk
Lancs L39 1QR
(telephone: 0695 79101)

Sinclair Research
Stanhope Road
Camberley
Surrey GU15 3PS
(telephone: 0276 685311)

Stonechip Electronics Ltd
Unit 9, Brook Industrial Estate
Deadbrook Lane
Aldershot
(telephone: 0252 318260)

Success Services (Pickard controller)
154 High Street
Bloxwich
Walsall
W Midlands WS13 3JT
(telephone: 0922 402403)

Thurnall Electronics
95 Liverpool Road
Cadishead
Manchester M30 58G
(telephone: 061 775 4461)

Transform
41 Keats House
Porchester Mead
Beckenham
Kent
(telephone: 01-658 6350)

U-Microcomputers (add-on expansion boards)
Long Lane
Warrington
Cheshire

William Stuart Systems
Quarley Down
Cholderton
Salisbury, Wiltshire
(telephone: 098064 235)

Technology Research
356 Westmount Road
London SE9 1NW
(telephone: 0784 63547)

Timedata (ZXM soundbox)
16 Hemmels
Laindon
Basildon
Essex SS15 6ED
(telephone: 0268 418121)

Trojan Products
166 Derlwyn Dunvant
Swansea SA2 7PF

Voltmace Ltd
Park Drive
Baldock, Herts
(telephone: 0462 894410)

Zeal Marketing Ltd (sound booster)
Vanguard Trading Estate,
Storforth Lane
Chesterfield S40 2TZ
(telephone: 0246 208555)

Index

THE FIRST RELIABLE GUIDE TO THE ADD-ON JUNGLE

Joysticks, keyboards, memory packs, microdrives, disk drives, speech units, sound synthesisers, light pens, digital tracers, modems, printers, and Input/Output devices – there are hundreds of add-on units available for the Sinclair Spectrum, at prices to tempt every user. But which are the best, which are the most useful, and which give you real value for money?

The Spectrum Add-on Guide answers these questions and many more. In easy, non-technical language it explains what each of the most important add-ons does, how it works, and most important of all, how you can actually use it in your programs. Whether you're a dedicated games player, a compulsive programmer, a serious business user or a budding author you'll find all the information you need to set up a tailor-made system and save the price of this book many times over. There are detailed program listings for the 'best buy' in each section, and a complete adventure game that can use up to seven different add-ons, including two networked Spectrums.

The Author

Allan Scott is a professional writer and editor and an award-winning audiovisual producer. He uses computers for graphics design, word processing, and data handling, and has built up a complete home system based on the Spectrum. *The Spectrum Add-on Guide* is the result of his experiences.

Front cover illustration by Alan Craddock

GRANADA PUBLISHING
Printed in Great Britain 0 246 12563 2

£5.95 net