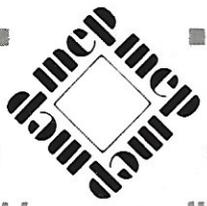
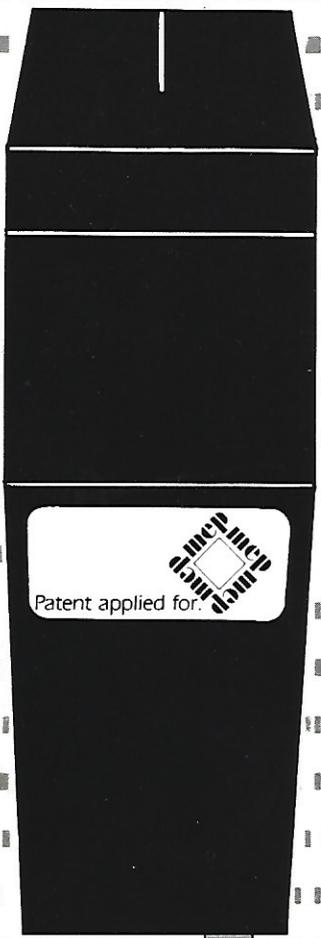


# THE BAR CODE READER PROJECT

**User Manual**

**George Hill**



Addison-Wesley Computing

# THE BAR CODE READER PROJECT

## User Manual

**George Hill**

Project Director:

**Mike Bostock**



**ADDISON-WESLEY PUBLISHERS LIMITED**

Wokingham, England. Reading, Massachusetts. Menlo Park, California  
Don Mills, Ontario. Amsterdam. Sydney. Singapore. Tokyo. Mexico City  
Bogota. Santiago. San Juan

### **Acknowledgements**

The Telepen system used by this bar code reader was originated by S.B. Electronic Systems Ltd., Arden Grove, Harpenden, Hertfordshire, AL5 4SL.

The publishers wish to thank the following organisations for their kind permission to reproduce materials in this manual:

Acorn User magazine, Redwood Publishing – 68 Long Acre, London WC2E 9JH for their tape-to-disk transfer program;

Beebug Publications Ltd, P.O. Box 109, High Wycombe, Buckinghamshire, HP10 8NP for the program 'Squaredance';

The Article Number Association (UK) Ltd, 6 Catherine Street, London WC2B 5JJ for information on the EAN coding system drawn from the ANA's Article Numbering and Symbol Marking – Operating Manual;

Lyons Tetley Ltd, 325 Oldfield Lane North, Greenford, Middlesex, UB6 0AZ for the photograph of one of their products.

### **Technical support**

#### **I M P O R T A N T**

Your bar code reader, although robust in construction, must be treated with the utmost care. It has an optical resolving capability of 0.1524 mm.

Where possible always use the acetate sheet provided, this will prevent paper dust and ink from building up in the reading tip.

**Do not use solvents of any type to clean any part of the reader.**

**The tip of the unit must be kept free from dust and dirt, etc., by the use of a soft brush.**

**Under no circumstances allow the unit to be dropped.**

If, having taken these precautions, the unit malfunctions please contact:

P.M.F. Design and Developments Ltd

"White Heather"

Noak Hill Road

Noak Hill

Romford

Essex, RM3 7LL

Telephone : 04023 48956

Please send any further technical queries concerning the bar code reader to the above address.

These materials have been developed by the Microelectronics Education Programme

© Text 1984 Council for Educational Technology

3 Devonshire Street, London W1

© Artwork 1984 Addison-Wesley Publishers Limited

Finchampstead Road, Wokingham, Berkshire

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

ISBN: 0 201 15472 2

Typeset by Quorum Technical Services Ltd.

Printed in Great Britain by Tecmedia

Photographs supplied by Format Photography, Newcastle-upon-Tyne

Cover design and illustrations by Jonathan Doney

ABCDEFGH 8987654

# Contents

## Section One Using the bar code reader pack

1	Introduction	1
2	How to use this manual	1
3	Using the bar code reader with your BBC microcomputer	2
4	Scanning technique (SCAN)	3
5	Reading bar code programs (BCBASIC)	5
6	Reading product codes (EAN)	7
7	Messages (DECODE)	11
8	Music (MUSIC)	13
9	Picture scanning (PICTURE)	13
10	Printing bar codes (PRINTER and TPUSERS)	15

## Section Two Worksheets

1	SCAN (Practising your technique)	19
2	BCBASIC (Bar codes and BASIC)	20
3	EAN (Reading EAN13 product codes)	21
4	DECODE (Bar code messages in binary form)	23
5	MUSIC (Music in binary form and the Musical Jigsaw)	25
6	PICTURE (A simple picture digitiser)	27
7	PRINTER (Bar code printout of BASIC programs)	28

## Section Three Program listings

29

## Section Four Appendices – background information

35

1	Tape-to-disk transfer with your BBC microcomputer	35
2	The Telepen system	35
3	The EAN13 system of product numbering	37
4	Decimal, binary and hexadecimal	39
5	Encoding messages in binary and bar code form	41
6	The music coding system	42
7	How the bar code reader works	43

# Foreword

This pack is designed to demonstrate Information Technology in action as exemplified by the applications of an optical input device to a microcomputer. The variety of activity provided in the pack should provide a valuable enhancement to any studies which wish to examine practical aspects of the New Technology. In addition the bar code reader should also offer much scope as a facility which could be used in a number of ways for the enhancement of learning.

This pack, together with its wealth of ideas, would not have happened but for the hard work, inspiration and encouragement provided by many people, some of whom deserve a special mention.

Foremost must be Richard Fothergill, MEP Director, who had the original idea for this pack and so started the whole thing off.

Special thanks must go to George Hill for his considerable talent in not only taking this handbook far beyond the original brief, but also in providing a substantial programming contribution.

Thanks must also go to Linden Thomas and Chris Hill of the North East Wales Institute for their original work on Telepen code and for providing some of the technically complex sections of program, and also to David Benzie for his original pioneering work on bar codes that provided much guidance for this work.

I am most grateful to Joe Telford for providing the cultural input to the pack in the form of the musical interlude, and to Anthony Lucas, project programmer, for his dedication in developing and finishing all the programs in the pack to their current high standard.

However, none of the pack would have been possible at all without the ingenuity of Paul Fuller of PMF Designs who produced the reader itself to an original design utilising the latest fibre optics technology.

Finally, thanks are due to the following and others not mentioned here whose inspired contributions have enhanced this project in one way or another.

John Coll  
Tony Quinn  
Noel Whalley  
Fay Winkworth  
Ray Lumb and Quorum Technical Services Limited  
The staff and pupils of Dinnington School

Mike Bostock  
Project Director

August 1984

---

# Section One **Using the bar code reader pack**

---

## **1 Introduction**

Bar code readers are not new devices but they have, until now, been beyond the reach of most schools and colleges because of their complexity and consequent high price. An additional disadvantage has been the specific nature of the programs available.

This bar code reader package has been put together by the Microelectronics Education Programme. It has several complementary aims. It is designed to marry technical innovation (the bar code reader itself) and complicated intermediate programming (the programs to translate the signals from the bar code reader into letters or numbers) but to be simple to use. The 'menu driven' approach has been adopted wherever possible to create a user-friendly environment.

It is first and foremost an educational package designed to illustrate some of the ways in which Optical Mark Reading (OMR) is used in industry and commerce. It has applications at almost all levels of computer education from elementary computer appreciation (in the form of a simple point-of-sale simulation, with little theoretical background) to Advanced Level (where it could be used as an illustration of interfacing, electronic communications and as the basis of an extended programming project).

Two standard systems of information coding are examined in detail. These are the Telepen data code, and the EAN13 product code. A third system is introduced, in which binary digits are encoded directly into bar code form. This may

be used at various levels as a teaching aid. It gives a good idea of how information is digitised and then converted into bar code form. It can also give valuable practical experience of the binary and hexadecimal systems. A stencil is provided so that original messages can be drawn using this system. For those with enough background knowledge it should also be possible to write tunes in bar code form with the stencil. Finally a very simple picture digitiser is included.

Another aim of the package is to provide a valuable labour-saving method of copying computer programs from books or magazines. These currently take the form of listings, requiring some skill as a typist and much patience to copy. Programs published in bar code form are able to be scanned in by a simple, entirely mechanical process. This is both reliable and quick and this technique will remain available as a useful alternative for the publication of programs on the printed page. The pack also provides a means of producing programs in bar code form using a suitable printer.

Once the bar code reader is in common use it is anticipated that other applications will rapidly be found for it, and the necessary programs written. These might include, for instance, a library management system and a stock control system.

We hope that you will find your bar code reader package interesting, informative and useful.

## **2 How to use this manual**

The manual is organized in four sections. The first section is intended mainly for teachers and is a detailed guide to the programs for 'trouble-shooting' purposes. The second section consists of very brief instructions on running the programs, and a few brief suggestions of

exercises or projects which might be undertaken by pupils using the program. The third section consists of examples of bar codes of various types. The final section contains detailed information on the coding systems being used and other important and appropriate information.

Teachers are encouraged to photo-copy appropriate pages of the second section and to use them as handouts for class or individual work. The bar codes may also be copied, and will be as 'machine-readable' as the originals if a good quality copier is available. Poor copies will of course lead to loss of readability. Generally, copies which are slightly too dark perform better than those which are too light, but there must be clear definition of the black/white boundaries.

Whenever bar codes are being read it is important to cover them with the clear acetate sheet provided. If you do not do this the codes

will be read correctly a few times but will eventually become polished. The light reflected from the polished surface prevents the bar code reader from distinguishing the black/white boundaries and impairs its performance. Eventually the code could become totally unreadable. This will also apply to copies of the bar codes as well as to the originals.

The acetate sheet is similar to that used for overhead projectors, so extra, or replacement sheets should be available in schools and colleges, or from a good stationery supplier. The sheets can be lightly polished to improve performance.

### 3 Using the bar code reader with your BBC microcomputer

#### INSTRUCTIONS FOR USE WITH THE BBC MICROCOMPUTER

The bar code reader pack should contain:

- the bar code reader (Figure 3.1);
- this instruction manual;
- a disk or tape;
- a piece of clear acetate sheet;
- the stencil.

The bar code reader is connected by a ribbon

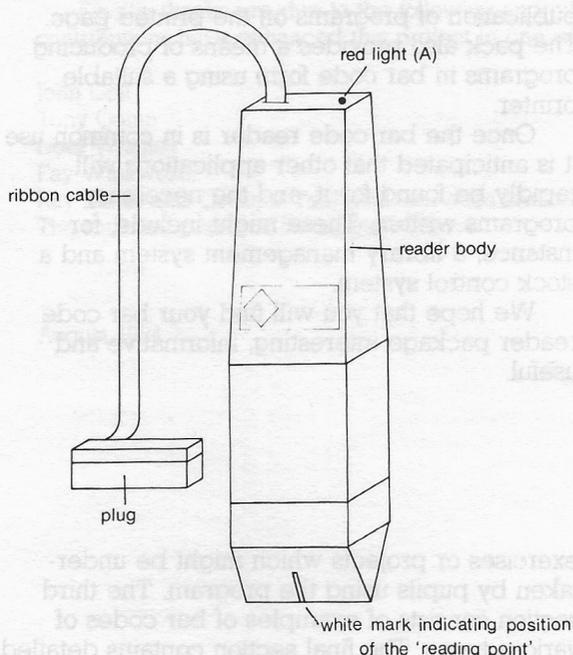


Fig. 3.1 The bar code reader

cable to a plug which fits the **user port** under the BBC microcomputer.

#### Connecting

1. Switch OFF the microcomputer.
2. Plug the bar code reader into the user port socket. Push aside the clips if necessary, and press the plug firmly until the clips 'click' home.
3. Switch the microcomputer back ON again.

#### The first trial

The red light **A** (see Figure 3.1) on the end of the bar code reader should be alight. If it is not, check that the bar code reader is correctly plugged in and that the microcomputer is switched on. If the light stays unlit and you have failures in subsequent trials, please return the unit for checking to the address given on page ii.

Now place the wedge-shaped end of the bar code reader **vertically** on a flat, clean piece of white paper. The light should go out.

#### Disconnecting

The bar code reader may be left connected when not in use without damage or side effects to the microcomputer. There will be a small extra current drain. If you need to disconnect, then push both clips away from the plug. This should lever the plug up out of the socket. Avoid twisting or levering, as these can damage the pins of the user port socket. Do not pull on the cable during the removal of the plug.

If you want to know how the bar code reader works, a brief summary of the principles is given in Appendix 7. It uses fibre-optic technology and the wedge shaped design has been adopted to give reliable results.

If all has gone well so far you should proceed to the next section.

### Using the disk version of the programs

The programs provided on disk are loaded and run through a 'menu'. To start the system up place the disk in DRIVE 0. Now type

```
<SHIFT-BREAK>
```

i.e. hold down the <SHIFT> key, press and release the <BREAK> key, then release the <SHIFT> key.

There may be occasions where this method is inappropriate. For example if the system is to be used over a network, or DRIVE 0 is not available. In these cases the menu can be obtained by switching to the appropriate drive using the \*DRIVE *n* command, and then typing

```
CHAIN"TITLE"<RETURN>
```

The menu program is very simple to operate. To load and run a program simply enter the number of the program (e.g. 1 for SCAN, 2 for BCBASIC, etc.) and press <RETURN>.

### Using the tape version of the programs

The programs appear on the tape in the order in which they appear in this manual, i.e.

- |            |            |
|------------|------------|
| 1. SCAN    | 5. MUSIC   |
| 2. BCBASIC | 6. PICTURE |
| 3. EAN     | 7. PRINTER |
| 4. DECODE  | 8. TPUSERS |

To load and run any of them it is necessary to locate the start of the program on the tape. Let us take EAN as an example. Type

```
CHAIN"EAN"<RETURN>
```

## 4 Scanning technique

### YOUR WORKSTATION

To use the bar code reader satisfactorily you will need a **firm flat surface**, about 50cm square. This should be within easy reach of the

The screen will show

### Searching

Run the tape forward, stopping it periodically and pressing the PLAY button. The screen will show the title of the program, followed by a number, e.g.

```
BCBASIC 06
```

Continue searching until you locate the end of the previous program. The screen will show, e.g.

```
BCBASIC 2B16
```

This time allow the tape to play on and the message

```
Loading  
EAN 00
```

should appear. It is now only a matter of waiting until the program load is complete. When the program starts to run, press the STOP button on the tape recorder.

If, after purchasing the tape version, you upgrade to disks, the method of transferring the programs to disk is detailed in Appendix 1. The programs TITLE and MENU are included at the end of the tape.

### GENERAL WARNINGS

The 6502 second processor alters the speed of response of the BBC microcomputer and the programs will *not* work correctly if the second processor is switched on. It is not necessary to disconnect the second processor, merely to ensure that it is switched OFF when bar codes are to be read and interpreted by the system.

The bar code reader uses the 6522 VIA chip and the performance of the bar code reader may be affected by any other device fitted to the BBC microcomputer which also uses the 6522. This may include some 'add-on boards'. The reverse is also true - i.e. the connecting of the bar code reader may affect the functioning of the other device.

keyboard and you should, for comfort, be able to see the VDU without having to turn your head. You will also require a 30cm ruler or other suitable straight edge and the piece of clear acetate sheet provided with this pack.

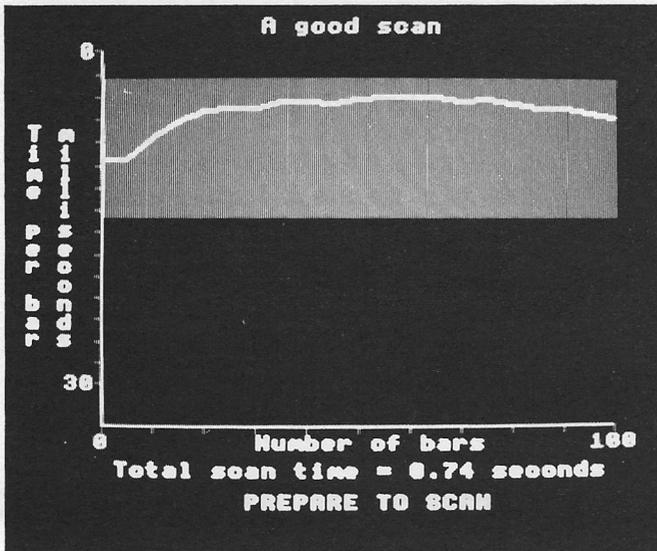


Fig. 4.1 The screen layout for the SCAN program



Fig. 4.2 Scanning bar codes using the bar code reader

Direct sunlight or very bright artificial light may impair the performance of the bar code reader and this should be avoided when the bar code reader is in use.

### GETTING THE SCANNING SPEED RIGHT

Successful use of the bar code reader does take a little practice. The program SCAN is designed to help you to develop a good scanning technique. You should ensure that you are achieving consistent success before you move on to the other programs.

Load and run the program SCAN

Axes will appear on the screen, and the message

### PREPARE TO SCAN

is seen at the bottom of the screen. The screen layout is illustrated in Figure 4.1.

Refer to Figure 4.2. This illustrates the best method of holding the bar code reader. You should now try scanning the tall, evenly spaced bars in Figure 4.3. Cover the bars with the piece of acetate sheet. This protects them from wear and stops the bar code reader tip from becoming clogged with paper dust. Place the

ruler along the centre of the bars, overlapping at both ends. Hold the bar code reader in a vertical position with the point of the wedge in contact with the paper and its long edge in contact with the ruler, as shown in Figure 4.2. The 'reading point' on the wedge should be clear of the first bar. The program will generate a high pitched beep and the message

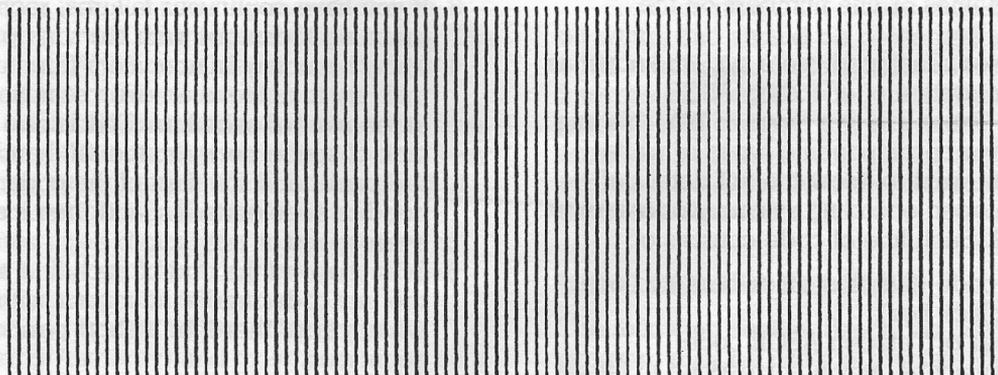
## SCAN NOW

will appear at the bottom of the screen. Keep the whole of the end of the bar code reader in contact with the page. Move the bar code reader over the bars at an even speed as if drawing a pencil line using the ruler as a straight edge until it is clear of the last bar. At the end of the scan lift the bar code reader from the paper.

The program will calculate your scan rate and graph it on the screen. Provided that your white line is within the red region and is reasonably straight then your scan was satisfactory. Messages will be printed at the top of the screen to report on your progress.

Practise several times and then repeat with the smaller bars in Figure 4.3. These are about the size which program codes will be. Place the ruler slightly below the line of bars. Try starting with the bar code reader tip close to the beginning of the bars and finishing close to the end with the bar code reader in contact with the paper. This is the technique which you will need to use to read programs in bar code form.

When you are confident that you and your bar code reader are performing well together move on to the next section.



Run the reader over the complete length of the bars



**Fig. 4.3** Bars for use with the SCAN program

## 5 Reading bar code programs

First plug in the bar code reader (if you have not already done so) following the instructions in Chapter 3. Now set up your workstation as described in Chapter 4 and if you have not practised your reading skill with SCAN, do so before proceeding any further.

### USING THE BCBASIC PROGRAM

Load and run the program BCBASIC

This is an assembly language program which

reorganizes the character input routines. After it is installed, BASIC will accept characters from the keyboard *and* from the bar code reader.

When BCBASIC has loaded and run an appropriate message will appear on the screen. The computer is now ready. Input may come from the bar code reader or from the keyboard. Between scans of bar code all the normal keys have their usual functions. This means that editing is possible by the normal means using the cursor keys and copy keys. Beware though!

If half-finished lines are left on screen while editing some odd results may occur.

[ BASIC programs are stored in the BBC computer in 'tokenised' form. Each keyword (e.g. PRINT, FOR, etc) is translated into a number. When accepting input from any source BASIC translates the words into these tokens. The computer recognises abbreviations for most words (e.g. P. for PRINT and N. for NEXT). It translates them into the same tokens as the full keywords. In order to reduce the amount of coding necessary the bar code programs are in fully abbreviated form and will appear initially on the screen with all possible shortenings. This may make them hard to read. However, after you have entered the *complete* program, LISTING it will give the normal fully spelled-out form.]

Each line of bar code represents a number of characters. These characters will rarely start and finish at the beginning or end of a line of BASIC program, so do not be disturbed to find half-finished lines or line numbers on the screen.

The codes presented use a system known as the Telepen system of encoding ASCII characters. There are two types of bar, thin and thick, and two corresponding types of white space. If you want to know the full details of how the Telepen encoding system works turn to Appendix 2. Each line of code consists of:

1. a **Start** code (five thin black bars and one broad black bar);
2. a number of Telepen ASCII code representations, which correspond to letters of the abbreviated BASIC program;
3. a checksum code to ensure that the line has been read correctly;
4. a **Stop** code (the reverse of the **Start** code).

To read in the short sample program (Program Listing 1) simply scan the lines of bar code in turn. You should start with the 'reading point' of the bar code reader as far to the left of the code as is practical without going off the edge of the paper. Use the ruler as a guide, placing it along the bottom of each line of bar code in turn, so that the 'reading point' is along the middle of the line of bars. Do not forget the piece of acetate sheet. This prevents damage to the codes.

A successful scan is rewarded with a high pitched sound, and the appearance of the text on the screen. A failure gets a low pitched beep. If a line fails to read correctly the first time, check the position of the ruler and scan it again. Try the list of tips at the end of the chapter if you cannot achieve successful scans.

The program may now be LISTED, RUN, SAVED, and edited in the normal way for any BASIC program. Before entering another program, type NEW to dispose of the old one. Some other programs are included in the Program listings for you to experiment with.

### Some tips gleaned from our experience

1. Ensure that the surface is flat. One of the best ways of ensuring this is to use several blank sheets of paper under the sheet containing the bar codes.
2. When starting a line scan, place the bar code reader on the paper in the correct starting position and ensure that the light on top of the reader has *gone out* before commencing the scan. To ensure this place the bar code reader on the page by putting it down quickly without 'bounce'.
3. If you have continuous failures, recheck your scanning technique with the SCAN program.
4. Many scanning failures stem from an inability to achieve an even scanning speed at the *start* of each line. This is due to the lack of space between the left hand edge of the paper and the start of the code. You need to accelerate fast and then maintain an even speed.
5. Scan *beyond* the end of each line and leave the bar code reader in contact with the white paper there for a short period after each scan.
6. Hold the bar code reader low down, with your fingers brushing the paper, and guide it along the ruler.
7. Some bar code is printed in a more compressed form having more, thinner bars per line. This will have to be scanned more slowly than the normal code.
8. Make sure that your piece of acetate sheet has not become damaged or very dirty. If it has, move it around until you find a clean portion. In the last resort, use a new piece.
9. Make sure that the bar code is of good quality and is without blemishes. Minor blemishes can sometimes be avoided by changing the level of the ruler, and reading a slightly different section of the strip of code.
10. If the code is printed on very thin paper, place a blank sheet of paper underneath to ensure that stray black marks are not being detected through the paper.
11. If the light is very bright move into shade or turn down the light.

## 6 Reading product codes

Many products such as groceries and books have bar codes on them. For instance, you might see them on the back of a new paperback book or on the bottom of a cereal packet. You will probably see one and maybe two bar code patterns there in a panel. An example of such a panel is shown in Figure 6.1. The taller of the two sets of code, with the numbers at the bottom is the one this program is designed to deal with. The code is in EAN form. This is the European Article Numbering code, one of the EEC's internationally agreed systems of coding. (The smaller code may be of many kinds and is not dealt with here, except to say that this particular code can represent the price of the article in pence.)

Two EAN codes are in common use on products. One is the EAN8 (eight digit) code which is for in-factory use by manufacturers. This program will *not* interpret these codes. The other is the EAN13 code which is a 13 digit code, and the numbers are reproduced in 'human readable' form under the bar code. The conversion of the numbers into bar code is very interesting, quite complex and is described in detail in Appendix 3. The first two and sometimes three digits refer to the country of origin or to the standardizing authority. Thus codes beginning with 50 belong to the UK standardizing system of the ANA (Article Numbering Authority) and those beginning with 978 (like those on the covers of some books) belong to the ISBN (International Standard Book

Numbering) system. There then follow ten digits (nine for the ISBN) which the authority may use as it pleases. The final digit is a 'checksum'. In the UK the first five of the ten 'national' digits refer to the manufacturer and the last five are an 'item number'.

The system does result in each item having a unique bar code printed on it. This code is used in a variety of ways in the wholesale and retail trades, and the program EAN is designed to give you some idea of how product codes are put to use.

It is important to note that true point-of-sale terminals are based on laser scanners. These devices can scan the code many times in a split second. This results in high reliability. The laser system can also cope with a range of colour combinations.

For best results the codes should be black on white, and should be presented to the bar code reader flat. Occasionally an incorrectly interpreted number will get through. In a real point-of-sale situation this is avoided by multiple scanning because a single scan may contain errors, but still produce the correct checksum digit (see Appendix 3).

### USING THE EAN PROGRAM

The following section describes how the program may be used to simulate various operations that a shopkeeper might perform using EAN13 product coding.

**Fig. 6.1** The EAN13 product code being scanned



## Scanning technique

The scanning method for EAN bar code is different from the 'ruler' technique recommended previously. Make sure the product panel is smooth and flat. Hold the bar code reader so that the wedge is parallel to the bars. Place it on the product panel to the left of the first bar. Sweep the reader smoothly across the code, freehand. The technique is illustrated in Figure 6.1. The program will know when the code has terminated (there are 'guard bars' in the pattern - see Appendix 3) and what you do at the end of the scan is not critical.

Load and run the program EAN

After the introductory screen (which you can pass over by pressing the space bar) the menu appears (see Figure 6.2). There are seven options which are chosen by pressing the appropriate number followed by the <RETURN> key.

### Throughout the program you can return to a menu by pressing the <ESCAPE> key.

In options 1, 2 and 6 below the same signals and messages are used. The messages appear at the bottom of the screen. They are

#### Scan now

indicating the computer's readiness to accept the next piece of EAN code or

#### Please scan again

indicating that an error has occurred. When you scan a piece of code a beep sounds immediately the scan is complete. If the scan was successfully decoded a high pitched second beep is heard and the EAN number will appear in the box on the screen. A low pitched second beep indicates an error.

### Option 1. Read codes

The simplest use of the code is to identify the country of origin of the article from its first two digits. Simply scan the code, and the product code should appear in the upper box, while the country of origin appears in the lower box. You may have to hunt around somewhat to discover articles which are *not* made in your own country.

### Option 2. Read and store codes

In this option you can store details of many different products in the computer's memory. Have a range of different items to hand - 10 or more to make the best use of the program.

For each item the procedure is:

1. Scan the code on the item. When this has been successfully accomplished, the flashing cursor will be seen in the box by the word **Manufacturer**;
2. Type in the manufacturer's name (or some suitable abbreviation if it is too long), terminated by <RETURN>;
3. The cursor will move to the **Product name** prompt. Type in the appropriate information, again terminated by <RETURN>;
4. Repeat this procedure for the **Size/Quantity** and **Price** prompts;
5. On typing the final <RETURN> the computer will be ready for the next item.

*Note:* You *must* make some sort of entry under each heading. If you do not know one of the first three items (e.g. the manufacturer's name) enter a space. If you do not know the price, option 6 will not work correctly on this item. You should enter 0 as the unknown price.

If you have made a mistake in the current entry press the <ESCAPE> key, and read on. Pressing the <ESCAPE> key returns you to a sub-menu which allows you to:

1. **Continue adding items** (this also deletes the current entry);
2. **Delete current entry**;
3. **Return to main menu**.

When you have completed the entry of all your items press <ESCAPE> and choose option 3 to return to main menu.

*It is strongly recommended that you now use option 4 from the main menu to SAVE the information you have just entered.*

### Option 3. Load file

*To use this option you must have prepared previously, on tape or disk, a file of codes and associated product information via options 2 and 4.*

If you are using disks, you *must* know the name of the file you wish to load and enter it at the prompt.

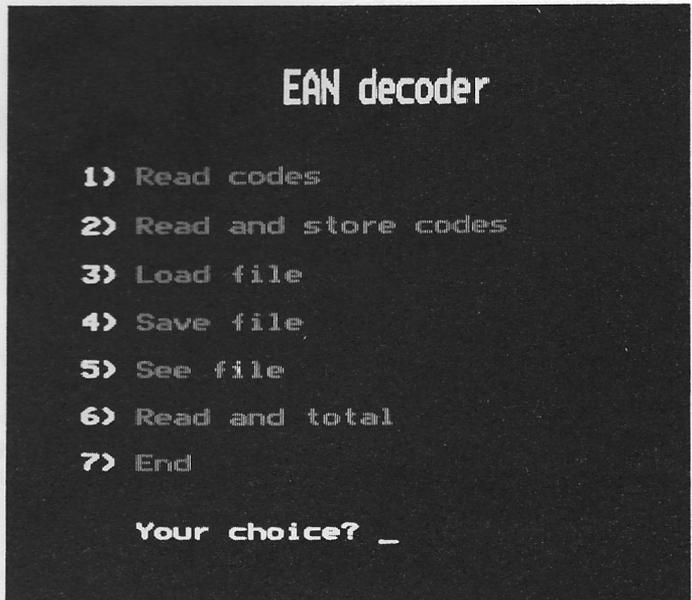
From tape if you enter a name the computer will search the tape for the named file. If you press <RETURN> without a filename the computer will attempt to load the next file on the tape, whatever it is. Note that errors may occur if you attempt to load, say, a BASIC program instead of a file.

After loading, the file becomes the current one for use in options 5 and 6. It may be

## THE EAN CODE READER AS A SIMULATED DECODER

A third way of using bar codes is to use them and black block bars to represent the binary digits 0 and 1. These binary digits are used to encode information concerning prices, quantities, etc. Information concerning EAN codes is in Appendix 1. The codes are read and interpreted by the EAN decoder.

**Fig. 6.2** The menu screen for the EAN program



extended by adding more items via option 2 (or option 6). After extension, do not forget to SAVE your file again via option 4.

Press <ESCAPE> to return to the menu.

### Option 4. Save file

In this option you may save information prepared in option 2 or extended via option 6 on tape or disk.

You have to supply a filename which must conform to the usual rules for program names detailed in the *User Guide* or *Disk Manual* for your particular machine. The file will be saved in the normal way and may be recalled via option 3 on a future occasion.

Press <ESCAPE> to return to the menu.

### Option 5. See file

This option allows you to 'browse' through the current file.

The cursor control keys (left and right arrows) allow you to step forwards or back through the items. The file 'wraps around', i.e. if you are at the end of the file and hit the right-arrow key you go to the start of the file and vice-versa.

Pressing the <SPACE BAR> will allow you to alter or update the information held under a particular EAN number. Pressing <ESCAPE> will take you out of this option and return you to the menu should you change your mind.

### Option 6. Read and total

This option allows you to simulate a point-of-sale operation. *To use it you MUST have a file which was prepared via option 2.* This may have just been prepared or may have been loaded in via option 3. You will also need some groceries or books whose details may or may not appear in the file.

Each time you scan an item which is in the current file its details appear in the box and its price is added to the total price. This is as it might happen at a supermarket checkout.

If you scan an item whose details are *not* found in the current file you will be prompted for the details and should type them in from the keyboard, exactly as in option 2. This item will be recognised in future during the current session.

If you make additions to the file during the current session and wish to SAVE the updated form of the file, wait until the end of the current 'checkout operation'.

Press <ESCAPE> to return to main menu. The current form of the file may now be saved via option 4.

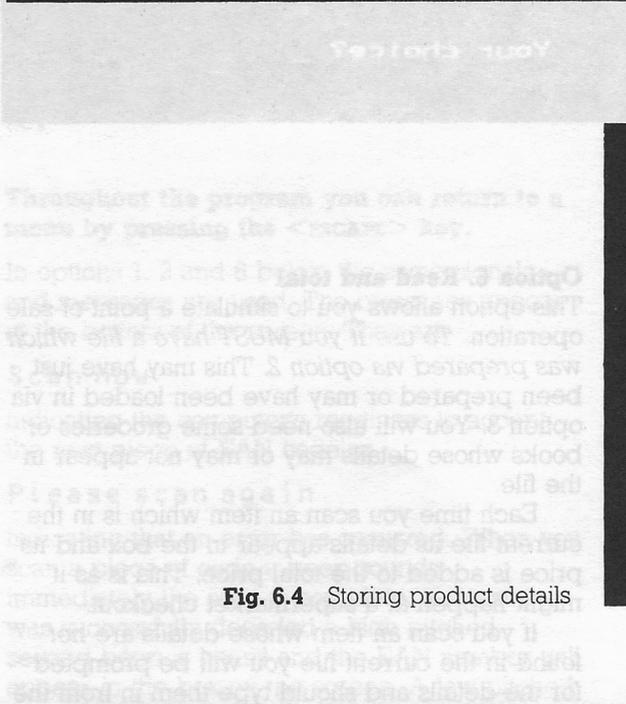
### Option 7. End

Type RUN <RETURN> to restart, if required.

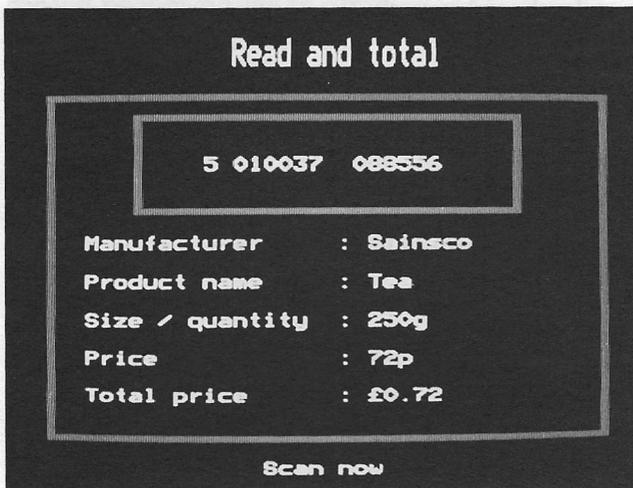
Figures 6.3, 6.4 and 6.5 show options 1, 2 and 6 in operation.



**Fig. 6.3** Reading product codes using the EAN program



**Fig. 6.4** Storing product details



**Fig. 6.5** Simulating a point-of-sale

# 7 Messages



Fig. 7.1 'Hello' in binary bar code

## THE BAR CODE READER AS A BINARY DECODER

A third way of using bar codes is to use narrow and broad black bars to represent the binary digits 0 and 1. These binary digits are used to encode letters which may be combined into messages. Information on the coding system is in Appendix 5. The `DECODE` program enables code of this type to be read and interpreted.

### USING THE DECODE PROGRAM

Load and run the program `DECODE`

After loading the menu will appear. To choose an option press its number followed by the `<RETURN>` key.

As a first trial read the message in Figure 7.1 using the following method:

1. choose option 1;
2. there is a checksum in the line so reply Y (OR YES);
3. scan the line using the technique described previously and the message should appear in the box.

The operation of program `DECODE` and the writing of coded messages is described now in more detail.

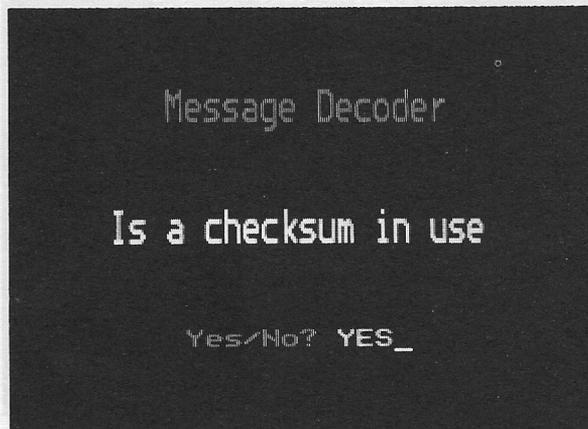


Fig. 7.2 Using a checksum?

Pressing the `<ESCAPE>` key returns you to the menu. Choose option 4 to exit from the program.

### Option 1. Scan and Decode

To use this option you need to know whether checksums are in use or not. They *are* in the examples given but may not be in codes you write yourself. Respond Y or YES, or N or NO as appropriate (see Figure 7.2).

In this option the message will appear in yellow inside the box on the screen after a correct scan. Up to eight lines of message can appear on the screen at the same time before the screen must be cleared. Instructions and information appear at the bottom of the box and messages about the validity of the scan appear at the top of the box. The messages at the bottom of the box should be self-explanatory.

There are two error messages.

**Scan error** indicates a general failure to recognise the patterns correctly. This may be due to an incorrect number of bars, incorrect scan rates, inaccurately drawn patterns or a variety of other causes.

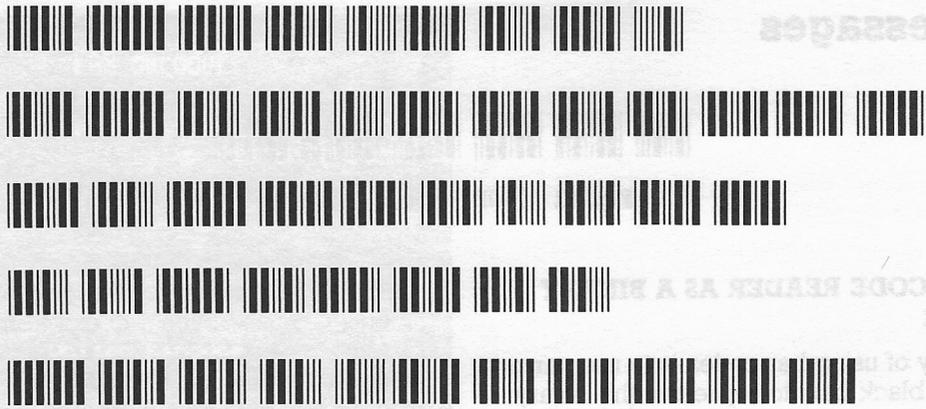
**Checksum error** indicates that the computer read the right number of bars but the last pattern did not give the correct checksum. This is normally corrected on repeating the scan but if you get continuous errors of this type it may be that the checksum digit is absent or incorrect. Exit to the menu, reselect option 1, respond N to the **Checksum?** question and try scanning again.

To assist you in this option 'beeps' are sounded. The middle pitched beep indicates that the computer is ready to accept the next scan. The high pitched beep indicates a good scan and should be followed by the appearance of the message. The low pitched beep indicates an error and will be accompanied by an error message.

Figures 7.1 and 7.3 contain messages for decoding by this and the following option.

### Option 2. Scan and Analyse

In this option the scanning technique, sounds and messages are very similar to option 1 though the display is different.



**Fig. 7.3** Your bar code reader should be handled with care.

Messages appear in the top section of the screen. The decoded message will appear one letter at a time in the central section of the screen (see Figure 7.4). The bar code equivalent of each letter will appear in the small box at the bottom of the screen exactly as it was seen by the bar code reader. Below it is its binary representation and beside it are the character and its ASCII code in both decimal and hexadecimal.

Pressing the <SPACE BAR> steps you along through the message one letter at a time to the end of the line when you are asked to scan the next line.

This section is to assist you to understand how the code is constructed before you use the

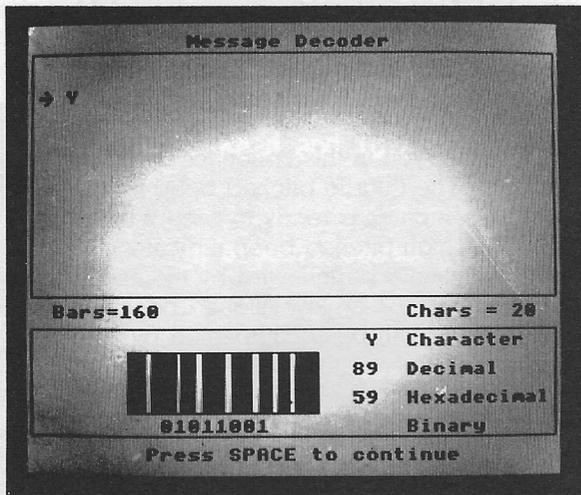
stencil and option 3 to construct your own messages.

### Option 3. Encoding Aid

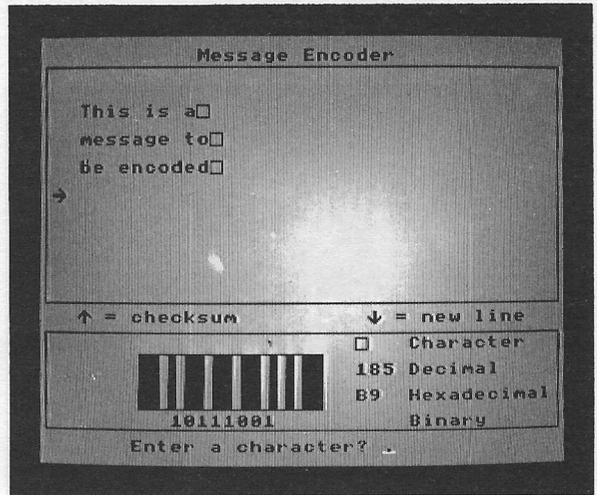
This option will help you to draw your own bar code messages.

You type your message one letter at a time. The character, its ASCII code in decimal, hexadecimal and binary and the corresponding bar code appear in the display. They are positioned as in option 2 (see Figure 7.5).

The codes can be drawn on a white piece of paper with an HB pencil. You can draw them using the stencil provided. Details of the use of the stencil are included in Appendix 5.



**Fig. 7.4** Scan and analyse



**Fig. 7.5** Encoding messages

## 8 Music

In this chapter, narrow and broad black bars are again used (as in the previous chapter) to represent binary digits. The information encoded represents musical notes. A brief explanation of the binary system can be found in Appendix 4 and details of the music coding system in Appendix 6.

To use this program no musical knowledge is needed. Some musical phrases are provided in bar code form among the example bar codes. There are two sets of phrases which comprise Tune 1 and Tune 2 (Program listings 6 and 7). These are referred to later in Worksheet 5. They are not named here to avoid giving away the secret and spoiling the Musical Jigsaw exercise in the worksheet.

### How music is encoded

Any musical note consists of two basic parameters. The first is how high or low the note is (its *pitch*). The second is how long it lasts (its *duration*). Both these parameters can easily be turned into numbers. In essence each 'byte' is divided into two portions. The bottom portion gives us a choice of 31 different pitches (or silence) while the top portion gives us a choice of eight different note durations. These codes are converted by the program into 'sound' statements and played by the microcomputer's sound generator. There is no volume variation, nor control of sound quality, though it would be possible to write a more complex music system which utilizes the bar code reader in this way.

### USING THE MUSIC PROGRAM

Load and run the program MUSIC

After loading, the menu is displayed offering 5 options (see Figure 8.1). There is a gap at the bottom of the screen. This is used for the display of the current activity or for error messages.

## 9 Picture scanning

### THE BAR CODE READER AS AN OPTICAL SAMPLING DEVICE

In this chapter the bar code reader is not used to read 'digital' information at all. It is used to

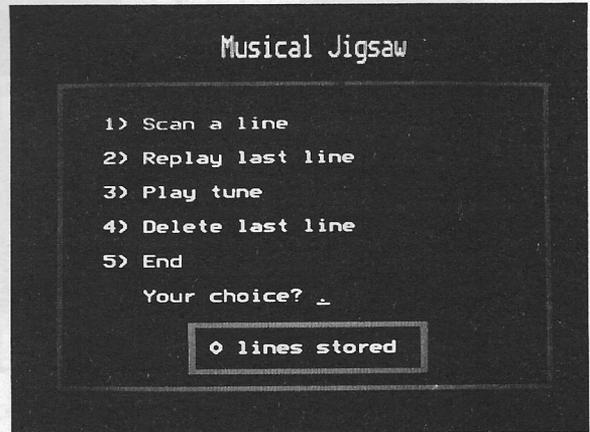


Fig. 8.1 Musical Jigsaw

You *must* select an option whenever the prompt is displayed. (See Figure 8.1.)

### Option 1. Scan a line

Any line of bar coded music can now be scanned. A successful read is rewarded with the playing of the line of music and the message at the bottom of the screen **Playing Current Line**. The two messages **Checksum error** and **Bad scan** indicate misreads of the code. Follow the advice — **Try again**.

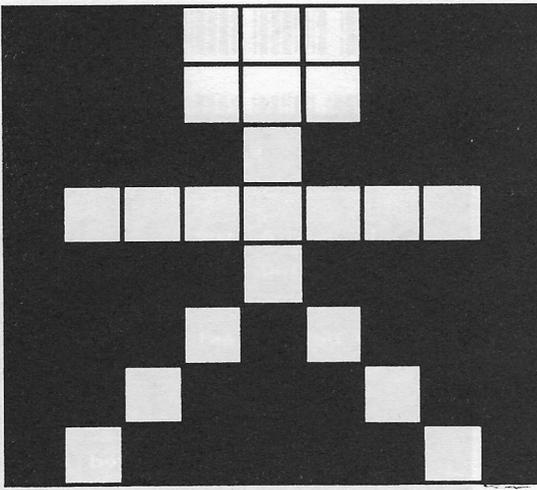
After each scan the line is automatically stored in memory and further lines can be added to the store.

### Options 2, 3 and 4

These options enable you to play the last line entered (2) or the whole tune (3). A rudimentary editing facility is included as option 4 which removes the last line from the store.

Press the <ESCAPE> key to return to the menu.

scan a picture. Pictures are essentially 'analogue' in nature. That is to say that the information on them is not divided up into small packets of information but is continuous. The bar code reader 'digitizes' the picture by scanning it



Use a ruler and scan each picture in horizontal strips

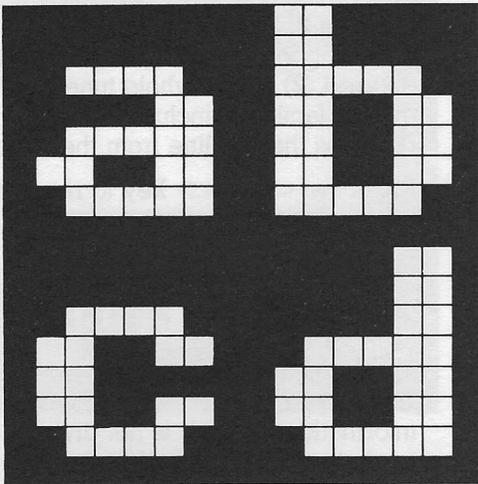
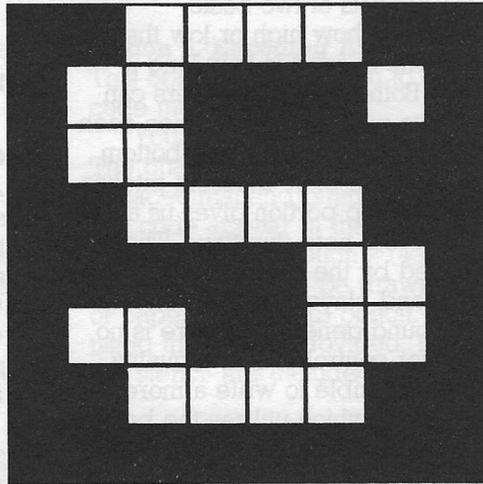


Fig. 9.1 Scanning pictures

in 8 or 16 horizontal sections but using many thousand vertical sections.

The bar code reader is not able to distinguish colours or shades of grey. It assumes a black and white world. As you sweep it across a picture (or a line of bar code) it samples the light intensity many thousands of times per second, converting black into 1 and white into 0. These 1's and 0's are now used to construct a screen image of the 'bar' which was scanned. The rate at which the light is sampled is fixed by the program so the user has to move the reader over the picture *at a constant speed* if distortion is to be avoided. The beginning of a line must be marked by a white/black change. The end will be assumed to be either a long section of white or running out of time.

## USING THE PICTURE PROGRAM

Load and run the program PICTURE

The screen turns white with black writing to be like the piece of paper. You have to choose

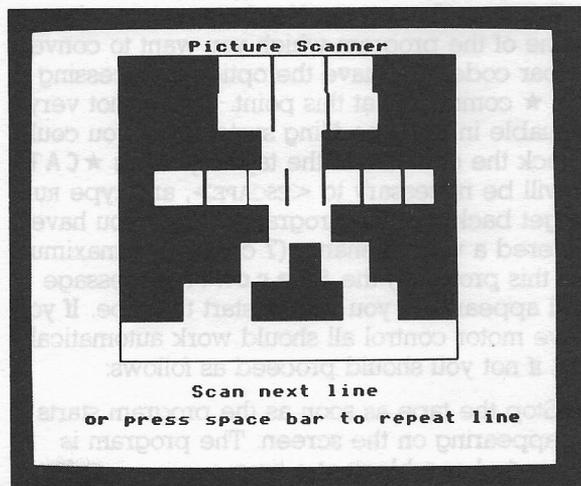


Fig. 9.2 A good picture scan!

## 10 Printing bar codes

### TO CONVERT BASIC PROGRAMS INTO BAR CODE FORM USING THE BBC MICROCOMPUTER

This program enables anyone with one of the printers mentioned below (and probably some others) to convert a normal BASIC program into the Telepen format suitable for reading by the BC BASIC system outlined in Chapter 5.

whether you want to scan 8 or 16 lines – type 1 or 2 in response to the prompt. (There are examples of pictures of both types in Figure 9.1.) After this initial choice the bar code reader will be ready to scan when you touch it on the paper. When you have the **Scan now** prompt showing at the bottom of the screen, scan the first line. If your scan was not too good – giving a distorted picture – it can be deleted by pressing the <SPACE BAR> and you can try again. Repeat this procedure until the picture is complete.

*Do not expect perfection.* Even the experts cannot produce an absolutely distortion-free picture. Figure 9.2 shows one of our better efforts!!

If you touch the bar code reader on white paper at the end of the last line, the program will terminate.

The program ends with the completion of the last line of the picture or the pressing of the <ESCAPE> key. This is indicated by the re-appearance of the prompt sign at the bottom left of the screen. If you wish to scan another picture after the program has ended, type RUN, followed by the <RETURN> key.

This program illustrates an important principle of picture digitization. Two things are needed to improve accuracy dramatically. The first is to make the speed of scanning absolutely constant and to take the same time to scan each line. The second is to scan more than sixteen lines. The more lines you can scan the better the picture will be (assuming that each line is accurate). This is clearly a case for a machine to do the scanning. A picture digitizer might have the bar code reader mounted on a frame above the page, so that it could be moved by motors at a constant speed across and in constant steps down the picture. The computer would store the results, perhaps in a disk file, for future processing.

The printers covered are those most common in home, educational and small business use including:

- Walters (two models);
- Epson MX, RX and FX series;
- Star 510, 515, Gemini and Delta;
- Shinwa (and other) CP80;
- Riteman;

- Canon PW1080A – and possibly other NLQ printers;
- Juki and Keyaki daisy-wheel.

The omission of a routine for the Seikosha GP100 is regretted but this was not possible for technical reasons.

The printing routines do *not* contain any ★FX5 calls. Parallel printers are selected by default (or ★FX5, 1). If you have a serial printer, then you must type ★FX5, 2 and set the baud rate with ★FX8, *n* to match that for your printer. (In certain cases it may be necessary to use ★FX156 with serial printers but you should know about this if it applies to you.) It will probably not be necessary to alter ★FX6 settings as all characters are sent to the printer via VDU1,c. Any ★FX call which is necessary must be issued either *before* the program is loaded and run or when you have access to the ★ commands when prompted by the program.

## USING THE PRINTER PROGRAM

Load and run the program PRINTER

A menu of printers to choose from is presented. You choose by typing the number of the appropriate printer on the menu.

The 'Epson' routine will work for any printer which has the 'ESC L n1 n2' double density graphics facility. This includes all Stars, CP80, Riteman, Canon and some others.

The 'Canon NLQ' routine produces narrower code than others, as this printer (and possibly other NLQ printers) produces more accurate bars than the others. This code has to be read at a slower speed than the conventional code but is more compact. (The CP80 also produces narrower code than the Epson and Star printers and this will need to be read more slowly and carefully.)

The daisy-wheel routine is applicable to most daisy-wheel printers as it uses the letter I as the basic bar.

You will be asked to set the printer to the top of a page, switch it off and switch it on again. This ensures that your listing starts neatly at the top of a page. For some of the options the 'skip-over-page-perforation' may be set at this point which avoids the possibility of the bars being printed straddling the perforations and being consequently difficult to read. If you have previously set this up yourself or the program is very short you can by-pass this feature.

*Note:* There are different sequences for tape and disk users.

In both cases, when the listing starts, the program name will be printed first. Then the program appears in abbreviated form on the screen. The line-number(s) of the line(s) contained in that piece of bar code are printed and finally the bar code itself.

*The lines of bars rarely correspond exactly to lines of program* but the line-numbers give you a check when reading the programs back in.

1 **Program from disk.** You have to state the name of the program which you want to convert to bar code. You have the option of accessing the ★ commands at this point. You can change drives (★DRIVE *n*), directories (★DIR *x*) or get a directory (★CAT or ★.) to check the contents of a disk. (BEWARE of commands like ★COMPACT and ★COPY. They will delete the program PRINTER from memory and it will be necessary to re-load it.) Once you have given a valid program name (7 characters maximum) printing of the bar codes should be completely automatic.

2 **Program from tape.** You have to state the name of the program which you want to convert to bar code. You have the option of accessing the ★ commands at this point. (This is not very valuable in the tape filing system, but you could check the contents of the tape by using ★CAT. It will be necessary to <ESCAPE>, and type RUN to get back into the program.) When you have entered a valid filename (7 characters maximum for this program) the **Searching** message will appear and you should start the tape. If you have motor control all should work automatically but if not you should proceed as follows:

- a) Stop the tape as soon as the program starts appearing on the screen. The program is loaded one block at a time;
- b) After the block has been printed the listing will stop;
- c) It is now necessary to rewind the tape a little and press PLAY. Ignore the **Block?** and **Rewind tape** error messages (you have just rewound the tape!). After the **Searching** message the **Loading** message indicates the correct block has been found and is being loaded. If **Loading** fails to appear, rewind the tape and try again.
- d) The next block will load, and you should repeat the steps from a) until the program has been completed.

If you find that this rewinding is irksome, or you are getting continuous errors, a utility TPUSERS is

provided to convert a BASIC program into a file for use with the PRINTER program. This will remove the **Block?** messages and the file will be loaded without the need for rewinding. A remote control facility makes this process much simpler.

The printing of program listings allows:

- the distribution of programs in a fairly indestructible form, cheaply through the post. There is no danger of accidental erasure by stray magnetic fields or customs X-ray machines;

- the production of backup copies of valuable programs. The sheets of bar code can be kept in a folder without danger of accidental erasure or deterioration.

In addition this is an ideal medium for distributing short, illustrative programs on the printed page as in a computer journal or user-group newsletter. Should a program have been sent out containing bugs, then a small slip containing the bar code amendments could easily be posted to users.

You will need a photocopier of a good quality.

- Use the broad laser beam first. You can try the effect of the best scanning technique in Chapter 4 or your teacher will demonstrate it to you. Scan the bars until your graph (the white bar) falls within the red area. Use a ruler to check.
- When you can do this, return on to the narrow laser beam and practice again.
- Finally try scanning the coded bands of the message. This means turning the reader so that it is parallel with the bars and scanning across the bars.

## QUESTIONS

1. Which scanning pattern is most reliable?
2. What are the mistakes you make using the narrow laser beam?
3. What are the longest and shortest times you can obtain, and still get the a good scan message?
4. Find the shortest time possible for a good scan. Count the number of bars in the sample (black + white). Calculate the average response speed of the program in bars per second.
5. Assume that the computer carries out about 50 operations to read each bar. How many operations per second is the computer performing?
6. As you scan the bars, pause briefly at the middle of the scan. Over how many bars did you travel too slowly to be accurate?
7. Scan the bars as they appear and then slowly but sweep up to the middle.
  - a) Between which bars was your performance greatest?
  - b) Choose two sections on the tape, each covering six bars. Work out the average speed of travel of the bar code reader over these sections in milliseconds per bar.
  - c) Measure the spacing of the bars and convert their speed into centimeters. Design an experiment where the bar code reader could be used to measure the acceleration of a wheeled object down a slope.

# Section Two Worksheets

Although this work and its ancillary parts are excluded from any photocopying or reproduction licensing schemes, the following section of Worksheets may be reproduced without payment of licence fee for use by pupils or students.

## USING THE PRINTER PROGRAM

Load and run the program: `PRINT`

A menu of printers to choose from is presented. You choose by typing the number of the appropriate printer on the menu.

The Epson routines will work for any printer which has the ESC L or ESC double density graphics facility. This includes all Star, CP80, Ritman, Canon and some others.

The Daisy NLC routine produces narrower code than others, as the printer (and possibly other NLC printed peripherals) more accurately hits the offset. The code has to be read at a slower speed than the conventional code but is more compact. (The CP80 also produces narrower code than the Epson and Star printers and it will need to be read more slowly and carefully.)

The daisy-wheel routine is applicable to most daisy wheel printers as it uses the same code as the basic Star.

You will be asked to set the position of the top of a page, which is 0 and refresh your again. This ensures that your page starts down at the top of a page. For some of the printers the skip-over page-percentage may be 0.00 at the point which avoids the carriage and the page being printed overwriting the page which is being consecutively printed. If you have previously set this to a value of 0.00 the program is very short. You may also use 0.00.

Note: There are also routines for dot-matrix type and dot-matrix.

2. Program flow tape. You have to state the name of the program which you want to control by bar code. You have the option of accessing the program directly at this point. (This is not very valuable in the tape filing system, but you could check the contents of the tape by using `*CAT`. It will be necessary to `<BACK>` and type the program name.) When you have entered a valid filename (7 characters maximum for this program) the Searching message will appear and you should start the tape. If you have prior control all should work automatically but if not you should proceed as follows:

- a) Stop the flow of code as the program starts appearing on the screen. The program is loaded and held at a block.
- b) After the block has been printed the screen will stop.
- c) It is now necessary to rewind the tape a little and press `RAW`. Ignore the Block T and Rewind tape error messages (you have not rewound the tape). After the Searching message the Loading message indicates the correct block has been found and is being loaded. If Loading fails or appears to stall the tape and try again.
- d) The next block will print and you should repeat the above steps until the program has been completed.

If you find that the routine does not work, or you are getting continuous errors, a further reason is

## Worksheet 1

## SCAN Practising your technique

Load and run the program SCAN

You will need a photocopy of Figure 4.3.

- Use the broad band of bars first. You can find details of the best scanning technique in Chapter 4, or your teacher will demonstrate it to you. Scan the bars until your graph (the white line) falls within the red area, four times out of five.
- When you can do this move on to the narrow set of bars, and practise again.
- Finally try scanning the broad band of bars freehand. This means turning the reader so that it is parallel with the bars, and scanning without the ruler.

### QUESTIONS

1. Which scanning method is most reliable?
2. What are the mistakes you make most often?
3. What are the longest and shortest times you can obtain, and still get the **a good scan** message?
4. Find the shortest time possible for a good scan. Count the number of bars in the sample (black+white). Calculate the maximum response speed of the program in bars per second.
5. Assume that the computer carries out about 50 operations to read each bar. How many operations per second is the computer performing?
6. As you scan the bars, pause briefly in the middle of the scan. Over how many bars did you travel too slowly to be measured?
7. Scan the bars so that you begin and end slowly but speed up in the middle.
  - a) Between which bars was your acceleration greatest?
  - b) Choose two sections on the graph each covering ten bars. Work out the average speed of travel of the bar code reader over these sections in milliseconds per bar.
  - c) Measure the spacing of the bars and convert these speeds into cm/second. Design an experiment where the bar code reader could be used to measure the acceleration of a wheeled object down a slope.

## Worksheet 2

## BCBASIC Bar codes and BASIC

Load and run the program BCBASIC

You will need a photocopy of Program listing 1.

- Cover the lines of bar code in Program listing 1 with the clear plastic sheet.
- Scan these lines of bar code in order. They form a simple BASIC program. When you perform a successful scan a high pitched beep sounds and the line of program appears on the screen. If you have made an error there will be a low pitched beep and you should re-adjust your ruler and scan again.

If you have continuous scanning problems refer to the hints listed in Chapter 5 or ask your teacher for help and advice. The most common problems are caused by not using a flat surface, uneven scanning speed and not holding the bar code reader vertical while scanning.

- When you have read in the whole program, type

```
RUN <RETURN>
```

and watch carefully what happens.

This program can be LISTED, and SAVED in the same way as any other BASIC program.

### ACTIVITIES

1. LIST the program (on a printer if you have one) and see if you can see the effect of each line, or section of the program.
2. SAVE the program on tape or disk.
3. Type NEW to dispose of the first program and read in one of the other examples (Program listings 2 and 3). RUN, LIST and SAVE it in the same way as before.
4. Read carefully the section on the Telepen coding system and try to answer the following questions.
  - a) How many frames are there on the first line of the program?
  - b) How many elements are there in each frame?
  - c) Draw the leader and trailer frames.
  - d) Which frame represents the checksum? Draw it.
  - e) What is the purpose of a checksum?
  - f) Find out the ASCII codes for the letters of your initials.
  - g) Work out the checksum for a line of Telepen code which represented your initials.
  - h) If you are feeling confident try drawing the frames of Telepen code for your initials and short messages.
5. What other uses do you think the Telepen system might be suitable for?

## Worksheet 3

# EAN Reading EAN13 product codes

Load and run the program EAN

You will need a selection of 13 digit bar codes (such as those found on many articles in the home).

After the title screen a menu of 7 options appears. Full details of these options are included in Chapter 6 but a summary is included below to help you. Most of the messages that appear on the screen do not require any explanation. Read the screen messages carefully at all times and try to help yourself before asking for help from your teacher.

Note that the scanning technique is not the same as for the other bar codes. The wedge should be placed on the white space in front of the bars, parallel to the bars. Now sweep the bar code reader smoothly and quickly over and past the bars in a single evenly-paced movement.

*(Throughout the program, you can return to the menu by pressing the <ESCAPE> key.)*

1. *Read codes* – This option allows you to read in product codes, and will tell you the country of origin of the product. If you read bar codes on books the term ISBN is shown instead of the country of origin. Find out what this means.
2. *Read and store codes* – This option allows you to read in the code for several products and type in additional information about them such as the name, price, weight, etc., of the product. The computer will remember this information, and you can use it in options 4, 5 and 6.  
Pressing the <ESCAPE> key returns you to a 'mini-menu' which allows you to correct mistakes, or exit to the main menu.  
After using this option and exiting to the main menu you should normally use option 4 to SAVE the information you have just typed in.
3. *Load file* – This option allows you to load a file of information from disk or tape if you have prepared and saved one on a previous occasion, using options 2 and 4. The file can be updated by option 2 or used in option 6.
4. *Save file* – This option is to save the information prepared by option 2 (or updated in option 6) on disk or tape. You have to supply a filename of up to 6 letters.  
Remember this name if you want to use the file on a future occasion.
5. *See file* – This option allows you to 'browse' through the file prepared in option 2 or loaded via option 3. The left and right arrow keys step you forwards and back through the file and the information is displayed on the screen. If you press the <SPACE BAR> you can alter the information relating to a particular code. Pressing the <ESCAPE> key will take you out of this option without alteration to the information.
6. *Read and total* – This option uses the information prepared by option 2 or loaded by option 3. When you read the product code, the details of the item will be displayed and a running total cost calculated.  
This option mimics the point-of-sale operation used in a supermarket.
7. *End*

## Worksheet 3 (continued)

### ACTIVITIES

1. Use option 1 to read some of your sample codes. Note down the countries of origin of these articles.
2. Use option 2 to build up a file of the articles whose codes you can read.
3. Look at your file using option 5, and then save it using option 4.
4. Exit from the program (option 7). Type RUN <RETURN> to restart it. Choose option 5 and try to see your file. There should be **No file to see**. Now use option 3 to load in your file and look at it again using option 5.
5. Use option 6 to imitate a supermarket checkout. After each article check that the computer has worked out the running total correctly. You can 'buy' articles more than once.
6. Find as many other articles which have 13 digit bar codes on them as you can. If possible cut out the bar codes, stick them on to a piece of paper and write the details (manufacturer, product, size, price) on the paper with the code.
7. Load your file using option 3 and then extend it to include your new articles using option 2. Save it using option 4 (you can replace the old file or give it a new name). You can now use this new file for a more extended point-of-sale exercise using option 6.

### QUESTIONS

Read the notes on the EAN system in Appendix 3 and answer the following.

1. What do the letters EAN stand for?
2. What does the 13 in EAN13 indicate?
3. What would EAN8 mean?
4. What are the first numbers of
  - a) an article originating in the UK?
  - b) an article originating in West Germany?
  - c) a book?
5. How many different sets of code are used?
6. How many fields are there?
7. How many guard patterns are there and where do they come? Draw them.
8. Take any EAN13 bar code. Draw it and mark on the drawing the guard patterns, the fields, which numbers are the country of origin and which number is the check-digit.
9. Under what circumstances are bar codes on products difficult to read using a bar code reader?
10. How do the large supermarkets using point-of-sale facilities cope with these problems?
11. How many manufacturers in total would the EAN13 codes be able to cover?
12. How many different products from all manufacturers could be referred to under the EAN13 system?

## Worksheet 4

# DECODE

## Bar code messages in binary form

Load and run the program DECODE

You will need photocopies of the messages given in Figures 7.1 and 7.3.

A menu of 4 options appears. Full details of these options are included in Chapter 7 but a summary is included below to help you. Most of the messages that appear on the screen do not require any explanation. Read the screen messages carefully at all times and try to help yourself before asking for help from your teacher.

*(Throughout the program, you can return to the menu by pressing the <ESCAPE> key.)*

You also need to know whether a checksum is being used. This is explained in Appendix 5.

1. *Scan and Decode* – This option allows you to read whole messages either from the bar code examples given or from messages you write yourself with the stencil.
2. *Scan and Analyse* – This option allows you to read in a line of message at a time and then step through it character by character. You do this by pressing the <SPACE BAR>. The display shows the bar code for each character and its ASCII code in decimal, hexadecimal and binary.
3. *Encoding aid* – This option allows you to type in a message, character by character. For each character the ASCII code is shown in decimal, hexadecimal and binary and the bar code is illustrated. More detail is included below in the section on using the stencil.
4. *End*

### USING THE STENCIL

When you use the Encoding aid option (above), the bar code for each character is illustrated. This can be formed, in two halves, using the stencil provided. If you do not understand binary, hexadecimal and decimal it does not matter but since this encoding is based on hexadecimal you may find it helpful to read Appendix 4 before going on.

The method is as follows:

1. Select option 3 – Encoding aid.
2. Type in the first letter of the message.
3. Look to the right of the bar code pattern. You will see the hexadecimal representation of the character. This is always two digits (or letters).
4. Using the stencil select the pattern for the first digit (the left one). If this is the first letter of the line draw it well to the left of the paper using an HB pencil. If it is not the first letter line up the stencil so that the right-hand dots of your previous pattern are showing through the holes to the left of the present pattern. You should fill in the bars well. Mark also the position of the two dots to the right of the pattern on the stencil.

## Worksheet 4 (continued)

5. Select the pattern for the second (right) digit. Line up the stencil so that the right-hand dots of your previous pattern are showing through the holes to the left of the present pattern. Fill in the bars and mark the positions of the right-hand holes. You may at this point space the next byte using the inter-stencil gaps from either the top or bottom lines.
6. Type in the next letter of your message and repeat the processes 3, 4 and 5 until you have only room for one more letter. Now press the < ↑ > key, read the pattern for the checksum and draw it as before.
7. Type in the next letter of the message and draw the next line.
8. Repeat until your message is complete.

You can omit the checksum altogether if you wish. Do *not* have checksums on some lines and not on others!

### ACTIVITIES

1. Use option 1 to read in some of the messages provided in the pack. These do use checksums.
2. Read Appendix 4 and answer the following questions.
  - a) How many symbols are there in binary, decimal and hexadecimal?
  - b) What do the letters ASCII stand for?
  - c) Convert 23 into binary.
  - d) Convert 237 into hexadecimal. (Use the computer or a calculator if you wish.)
  - e) Convert A3 (hex) into decimal. (Use the computer or a calculator if you wish.)
  - f) Convert D9 (hex) into binary. (Use Table 6 in Appendix 4.)
3. Read Appendix 5 and answer the following questions.
  - a) What bars represent binary 1 and 0?
  - b) What is a checksum?
  - c) What is the ASCII code for the first letter of your surname?
  - d) Either convert it into hex and binary by yourself or use option 3 (the encoding aid) to do it for you.
  - e) Draw it using the stencil.
  - f) Read it using option 1 or 2 (no checksum).
4. Write a message to someone using the encoding aid and the stencil:
  - a) without checksums;
  - b) with checksums.Which is read more reliably?

## Worksheet 5

# MUSIC Music in binary form and the Musical Jigsaw

Load and run the program MUSIC

You will need photocopies of Tunes 1 and 2 (Program listings 6 and 7).

A menu of 4 options appears. When the full-stop prompt appears you must choose an option by typing in its number. Otherwise the program needs no further explanation. Read the screen display carefully before asking for help or referring to Chapter 8 of the manual.

The music coding system is fully explained in Appendix 6. It is not necessary to understand this to do the first exercises.

### **THE MUSICAL JIGSAW**

Many common tunes consist of short sections of music (called phrases) which are arranged in a particular order to make the melody. Often phrases appear more than once in a tune.

Tune 1 consists of four phrases labelled A, B, C and D.

Scan them to hear what they are like.

Now delete them from the store (option 4) and try to construct a well known tune from the pieces. This tune contains eight phrases altogether, which might be for instance,

B B A C D A B D – (but they aren't!)

If you find it impossible to work out the answer is printed on page 42.

Tune 2 also has four phrases, A, B, C and D.

It has eight phrases in the finished tune with the last four repeated for the chorus. See if you can work it out. If not the answer is on page 42.

Can you name the tunes?

You can have some fun in scanning the lines in the wrong order, and seeing whether you can improve on the original!

### **MORE ADVANCED ACTIVITIES**

If you want to construct your own bar code music, then some knowledge of musical notation will be necessary and the detail of the coding is to be found in Appendix 6.

An example of a simple piece of encoding using the stencil is given there. Do you recognise the tune?

# Worksheet 5 (continued)

As a first exercise try encoding the following sequence of notes.

- G crotchet
- G crotchet
- A crotchet
- F# dotted crotchet
- G quaver
- A crotchet

Don't forget the checksum.

If you have the patience try to complete the tune by writing the next six phrases.

If your musical ability stretches that far, write your own jigsaws. Two appropriate tunes to use would be 'Frere Jacques' and the 'Ode to Joy' from Beethoven's Ninth Symphony.

## MORE ADVANCED ACTIVITIES

If you want to construct your own bar code music, then some knowledge of musical notation will be necessary and the detail of the coding is to be found in Appendix 6. An example of encoding using the stencil is given there. Do you recognise the tunes?

## Worksheet 6

# PICTURE

## A simple picture digitiser

Load and run the program PICTURE

You will need a photocopy of Figure 9.1.

The screen turns white with black writing, to be like the piece of paper. You have to choose whether you want to scan 8 or 16 lines – type 1 or 2 in response to the prompt. (There are examples of pictures of both types.) The bar code reader will be ready to scan when you touch it on the paper.

When you have the **Scan now** prompt showing at the bottom of the screen scan the first line. If your scan was not too good – giving a distorted picture – it can be deleted by pressing the <SPACE BAR> and you can try again. Repeat this procedure until the picture is complete.

*Do not expect perfection.* Even the experts cannot produce an absolutely distortion-free picture. Figure 9.2 shows one of our better efforts!

If you touch the bar code reader on white paper at the end of the last line or press the <ESCAPE> key the program will stop.

To scan another picture type

RUN <RETURN>

The bar code reader is neither able to distinguish colours nor even shades of grey. It assumes a black and white world. As you sweep it across a picture (or a line of bar code) it samples the light intensity many thousands of times per second converting black into 1 and white into 0. These 1s and 0s are now used to construct a screen image of the 'bar' which was scanned.

The rate at which the light is sampled is fixed by the program, so you have to move the reader over the picture at a *constant speed* if distortion is to be avoided.

### ACTIVITIES

1. First scan the example pictures until you are getting consistent results.
2. Draw your own picture with an HB pencil on squared paper and read it in.
3. Now try 'reading' a cartoon from a newspaper or comic. For the best results scan 16 lines and ensure that the picture is 'framed' by a black line on the left and a black line, followed by a strip of white on the right. If you need to add these two 'rails' on either side of the picture you should use pencil because biro ink will not be 'seen' by the bar code reader. It may help to divide the picture into 16 strips with ruler and pencil before scanning it.
4. Suggest some things which would make the system give better results with real pictures.

# Worksheet 7

# PRINTER Bar code printout of BASIC programs

To use this program you follow the prompts on the screen. Full details are in Chapter 10 of the manual.

You can only use this program if you have one of the printers listed in Chapter 10 or one which behaves similarly.

## ACTIVITIES

1. Write a short BASIC program and save it on disk.
2. Load and run the program PRINTER
3. Follow the prompts and get the program to print your program in bar code form.
4. Load and run the program BCBASIC and read your program back in using the bar code reader.

---

# Section Three **Program listings**

---

Although this work and its ancillary parts are excluded from any photocopying or reproduction licensing schemes, the following section of Program listings may be reproduced without payment of licence fee for use by pupils or students.

1MO.7 2X=70:REP.

3P.CHR.(129+RND(

5))CHR.141"MEP Add

ison-Wesley bar co

de reader" 4SO.1,-

12,RND(200),2 5X=X

-1:U.X=0

**Program listing 1** Test (BCBASIC)

1MO.1 2V.23,1,0;0

;0;0;29,640;440; 3

X%=0:Y%=0:D%=RND(5

0):E%=RND(50) 4D%=

RND(50):E%=RND(50)

:GC.3,RND(3) 5REP.

6MOV. X%,Y%:DR.-X

%,Y%:DR.-X%,-Y%:DR

.X%,-Y%:DR. X%,Y%

7X%=X%+D%:Y%=Y%+E%

8IF ABS X%>632 D%

=-D%:GC.3,RND(3) 9

IF ABS Y%>440 E%=-

E%:V.19,RND(3),RND

(7);0; 10U. INKEY(

-99) 11RUN

Program listing 2 Squaredance (BCBASIC)

1REP.:REA.A 2IFA=

0TH.U.A=0:END 3SO.

1,-15,A,4 4G=INKE

Y(20) 5U.A=0:END 6

D. 101,109,117,129

,121,121,137,129,1

29,149,145,149 7D.

129,117,101,109,1

17,121,129,137,129

,121,117,109 8D. 1

17,101,97,101,109,

81,97,109,121,117,

109,117,101 9D. 10

9,117,129,121,121,

137,129,129,149,14

5,149,129 10D. 117

,101,109,117,89,12

9,121,117,109,101,

81,101,97,101,0

1REM TAPE TO DISC

TRANSFER 2REM fro

m ACORN USER April

1983 3MO.7 4P.TA

B(5,10);"Press KEY

f0 to start trans

fer";TAB(5,11); 5\*

KEY0 \*TAPE|M\*FX138

,0,128|MCLS|MLO.""

||MA\$="" :I%=HIM.+20

0:J%=0:REP.A\$=A\$+C

HR\$(J%?I%):J%=J%+1

:U.I%?J%=32 OR J%=

7|M\*D.||MSAVE A\$|M

6END

Program listing 4 T\_D (BCBASIC)

# Section Four Background Information

1MO.7 2V.23,1,0;0



;0;0; 3REP. 4P.TAB



(RND(38),RND(21));



CHR.(128+RND(7));C



HR.(64+RND(26)) 5U



.0



**Program listing 5** Alpha (BCBASIC)



**Program listing 6** Tune 1 (MUSIC)



**Program listing 7** Tune 2 (MUSIC)



**Program listing 8** Message 3 (DECODE)

---

# Section Four **Appendices – background information**

---

These appendices are included to provide background information of a more technical nature on the bar code reader and the programs included in this pack.

## 1 **Tape-to-disk transfer with your BBC microcomputer**

To transfer the programs to disk there is a program in bar code form called `T_D` in the Program Listings section.

1. First check the disk you wish to transfer to. There must be at least 10 catalogue entries free, and it would be a good idea to **★COMPACT** the disk if it already contains programs or files. Better still use a new blank, formatted disk.
2. Now read in the `T_D` program in the way described in Chapter 5.
3. **SAVE** the program on disk by typing

```
★DISK  
SAVE "T_D" <RETURN>
```

4. Place the tape in the tape recorder, and rewind it. Press `<CTRL-BREAK>` to reset the system. Type

```
CHAIN"T_D"
```

and then press the `PLAY` button when the **Searching** message appears. The programs will transfer to disk in sequence.

Should any tape reading errors occur, rewind the tape to the beginning of the program

where the error occurred, using **★CAT** to locate it. Press red function key `f0` and transfer should continue.

As an alternative the programs can be transferred by the standard method: i.e.

```
★TAPE  
LOAD"filename"  
★DISK  
SAVE"filename"
```

The last program on the tape is `MENU` which can be run to select the bar code programs conveniently.

To enable the disk to 'autoboot' type

```
★OPT4,3 <RETURN>  
★BUILD !BOOT <RETURN>
```

When the figure 1 appears on the screen type **CHAIN"TITLE" <RETURN>**

When the figure 2 appears press the `<ESCAPE>` key.

The disk will now automatically run the menu program when you press `<SHIFT-BREAK>`.

## 2 **The Telepen system**

The Telepen system is a method of encoding ASCII characters into bar code.

The two essential features of the code are:

1. each character is represented by a 'frame' or bar code which is of the same total width;
2. the 'frames' are made up of 'elements' which consist of a black bar followed by a white bar.

The black bars are of two types, one of which is approximately three times as wide as the other,

though the absolute width does not matter greatly as long as the detector can distinguish them clearly. Each type of black bar has its white equivalent. This gives four possible combinations of black followed by white bar. Broad black shall be represented by 'B', narrow black by 'b', and broad and narrow white by 'W' and 'w' respectively. The four combinations are numbered 0 to 3.

- Element 0 — Bw
- Element 1 — bw
- Element 2 — BW
- Element 3 — bW

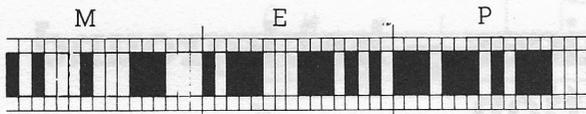


Fig. A2.1

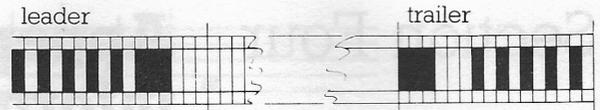


Fig. A2.2

The ASCII codes are numbers between 0 and 127 which can be represented as a seven bit (digit) binary number. (If you are not familiar with the binary system refer to Appendix 4.) Each of the elements 0 to 3 described above is associated with a particular pattern of 0's and 1's in the binary equivalent of the ASCII code. The method of encoding is somewhat complex though!

You first need to understand 'parity'. The parity of a number depends on whether its binary form contains an odd or even number of 1's. The particular type of parity used in Telepen is even parity. This means that if there are 1, 3, 5 or 7 1's in the number it has a 1 added to it as its 'parity bit', whereas if it has 0, 2, 4, or 6 1's in it a 0 is added as its 'parity bit', e.g.

0010101 — 3 1's thus parity bit = 1,  
coded as 10010101  
0111010 — 4 1's thus parity bit = 0,  
coded as 00111010

The resulting numbers always contain an even number of ones — hence **even parity**.

To make matters more awkward the binary digits increase in value from right to left but the Telepen scan runs from left to right and so before encoding starts we have to reverse the binary number! The various elements are associated with 'bit patterns' (i.e. patterns of 1's and 0's) as in Table 1.

Table 1

Element	Bars	Bit-pattern
0	Bw	00
1	bw	1
2	BW	010
3	bW	01 or 10 (see note)

Note: Element 3 always occurs in pairs in any frame. The first occurrence represents 01, the second 10.

Let us follow the encoding of the letters MEP into Telepen patterns. This is shown in Table 2.

Table 2

	ASCII	Binary	Parity	Binary	Reverse
<b>M</b>	77	1001101	0	01001101	10110010
<b>E</b>	69	1000101	1	11000101	10100011
<b>P</b>	80	1010000	0	01010000	00001010

The last patterns in Table 2 are now encoded as follows.

10110010 = 1 01 10 010 giving bwbWbWBW  
10100011 = 1 010 00 1 1 giving bwBWBwbwbw  
00001010 = 00 00 1 010 giving BwBwbwBW

If the capital letters are counted as 3 units and the small letters as 1 unit (the bar widths) it can be seen that the total width of each frame is 16 units.

bwbWbWBW  
= 1+1+1+3+1+3+3+3 = 16  
bwBWBwbwbw  
= 1+1+3+3+3+1+1+1+1+1 = 16  
BwBwbwBW  
= 3+1+3+1+1+1+3+3 = 16

The Telepen equivalent of MEP is shown in Figure A2.1.

To give the reading system time to 'settle down' a 'leader frame' is added at the start of each line and a 'trailer frame' at the end of each line. These are always as shown in Figure A2.2.

The final stage is the addition of a 'checksum' as the penultimate frame on each line. This is curiously calculated as follows:



Fig. A2.3

1. Add up the ASCII codes on the line (for MEP this is  $77+69+80=226$ )
2. Calculate the remainder when this number is divided by 127 (for MEP this is  $226 \div 127=99$ )
3. Subtract this number from 127 (for MEP this is  $127-99=28$ )
4. Convert this to Telepen code as described above.

28 becomes:

- 0011100 (in binary)
  - 10011100 (add parity bit to make 'even')
  - 00111001 (reverse)
  - 00 1 1 1 00 1 (sort bit-patterns)
- BwbwbwbwBwbw (code patterns)

The final form of the bar code line will be as shown in Figure A2.3.

### 3 The EAN13 system of product numbering

The EAN13 system represents thirteen decimal digits in bar code form. The code for the 13 digits is carried in two 'fields'.

The bar code is started by a 'guard pattern' of three narrow bars, two black and one white. This is followed by the code for the first field. The fields are separated by another guard pattern followed by the code for the second field. Another guard pattern terminates the code. The first field of 5 digits is the manufacturer's number and the second field of 5 digits is the item reference number. These divisions are shown in Figure A3.1.

There are three separate ways of representing the numbers from 0 to 9. These are shown in Figure A3.2. The important things to notice are:

1. In set A all the codes have either 3 or 5 black bars in total while sets B and C have either 2 or 4 black bars. Set A is referred to as an 'odd parity' set while sets B and C are 'even parity' sets;
2. Sets A and B always start with a white bar whereas set C always starts with a black bar. (Set C is the exact reverse of set B);

3. The pattern for every digit is the same total width of seven narrow bars.

The first field of the bar code is made up of patterns from sets A and B, while the second field is made up of patterns from set C. The fields are started, separated and ended by guard patterns (shown in Figure A3.3).

The second field is the simpler one. It represents the last six digits of the code in a straightforward manner using set C.

In the first field the first digit is represented by a pattern of six letters as below.

- 0 — AAAAAA
- 1 — AABABB
- 2 — AABBBAB
- 3 — AABBBBA
- 4 — ABAABB
- 5 — ABBAAB
- 6 — ABBBAA
- 7 — ABABAB
- 8 — ABABBA
- 9 — ABBABA

These patterns of letters are now used to select the number set from which the next six digit codes of the pattern will be chosen. Thus the UK codes consist of 15 patterns of bar code. These are arranged as shown in Table 3.

**Table 3**

- |                                      |   |
|--------------------------------------|---|
| 1 left guard                         | } - representing 5 - the first UK digit |
| 2 digit 2 pattern from set A         |   |
| 3 digit 3 pattern from set B         |   |
| 4 digit 4 pattern from set B         |   |
| 5 digit 5 pattern from set A         |   |
| 6 digit 6 pattern from set A         |   |
| 7 digit 7 pattern from set B         |   |
| 8 central guard                      |   |
| 9 digit 8 pattern from set C         |   |
| 10 digit 9 pattern from set C        |   |
| 11 digit 10 pattern from set C       |   |
| 12 digit 11 pattern from set C       |   |
| 13 digit 12 pattern from set C       |   |
| 14 checksum digit pattern from set C |   |
| 15 right guard                       |   |



**Fig. A3.1** The coding divisions for an EAN13 product code

VALUE OF CHARACTER	NUMBER SET A (odd)	NUMBER SET B (even)	NUMBER SET C (even)
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			

Fig. A3.2 Coding of number characters

Normal guard pattern  
(right and left)



Centre pattern



Fig. A3.3 Coding of guard patterns

Prefix Values			
00 - 09	(Reserved for UPC)	70	(Norway)
20 - 29	In-Store Numbers	73	Swedish EAN Committee (Sweden)
30 - 37	Gencod (France)	76	Schweizerische Artikelkode Vereinigung (Switzerland)
40 - 43	CCG (Germany)	77	APNA Australia
49	Distribution Code Center (Japan)	80 - 83	(Italy)
50	ANA (United Kingdom)	84	AECOC (Spain)
54	ICODIF (Belgium)	87	UAC (Netherlands)
57	Dansk Varekode Administration (Denmark)	90 - 91	BAN - Austria
61 - 62	(Reserved for DCI)	978	ISBN
64	The Central Chamber of Commerce (Finland)	979	Reserved for ISBN
65 - 69	(Reserved for DCI)	98 - 99	Coupon Numbers

**Fig. A3.4** Assignment of prefix digits by EAN

Note: Digit 2 is always 0 for the UK pattern. The total width of the bar code is always 95 units, made up from:

- 12 × 7 units for the 12 digits;
- + 2 × 3 units for left and right guards;
- + 1 × 5 units for centre guard.

The checksum digit is calculated by a complex algorithm. To follow it we need to number the 13 digits from left to right as digits 1 to 13. Thus the UK's digits 50 are digits 1 and 2 respectively. During the calculation of the checksum the checksum digit (number 13) is not used!

- Step 1 Add together digits 2, 4, 6, 8, 10 and 12 and multiply the result by 3.  
 Step 2 Add together digits 1, 3, 5, 7, 9 and 11.  
 Step 3 Add together the results of steps 1 and 2.

Step 4 Take the final decimal digit only of this sum, and subtract it from 10. The result is the checksum.

The calculation for the EAN number 50-00127-061-9-C (where C is the missing checksum)

- $(0+0+2+0+1+9) \times 3 = 12 \times 3 = 36$
- $(5+0+1+7+6+0) = 19$
- $36 + 19 = 55$
- $10 - 5 = 5 = C$

The fully encoded number is thus

50-00127-06109-5

This is shown encoded as the pattern in Figure A3.1.

Finally, for completeness a list of the meanings of the EAN prefix digits is given in Figure A3.4.

## 4 Decimal, binary and hexadecimal

This appendix only deals with whole numbers.

Our normal numbers are in **decimal**. The number is represented by several **digits**. Each digit is represented by one of the ten **symbols** 0123456789.

In a decimal number the position of a digit tells you its value. The right-most symbol is the units digit, the next to the left is the tens digit, the next is the hundreds and so on.

In **binary** there are only two symbols 0 and

1. The position of the symbol again tells you its value. The right hand end digit is again units. The next to the left is twos, the next is fours, the next eights, and so on.

In many applications binary numbers are grouped in sets of eight **bits** (Binary digITS). This group of eight bits is called a **byte**.

Here is a typical byte.

11001001

If this is translated into decimal then working from the right there is:

- $1 \times 1$
- $0 \times 2$
- $0 \times 4$
- $1 \times 8$
- $0 \times 16$
- $0 \times 32$
- $1 \times 64$
- $1 \times 128$

Adding these up the byte represents

$$1+8+64+128=201 \text{ decimal.}$$

The minimum value a byte can represent is when all the bits are 0. The maximum value is when all the bits are 1. This is the number

$$1+2+4+8+16+32+64+128=255 \text{ decimal.}$$

Decimal can be translated into binary by repeatedly dividing by two and looking at the remainder. Table 4 shows how to convert 153 decimal into binary.

**Table 4**

Step	Operation	Quotient	Remainder	BIT
1	153/2	76	1	units
2	76/2	38	0	twos
3	38/2	19	0	fours
4	19/2	8	1	eights
5	8/2	4	0	sixteens
6	4/2	2	0	thirty-twos
7	2/2	1	0	sixty-fours
8	1/2	0	1	128's

Thus 153 decimal is 10001001 binary.

It is clearly rather awkward to have to use decimal numbers to represent binary and vice-versa. Table 5 shows successive binary bytes counting up from zero.

**Table 5**

Binary	Decimal	Binary	Decimal
00000000	0	00001001	9
00000001	1	00001010	10
00000010	2	00001011	11
00000011	3	00001100	12
00000100	4	00001101	13
00000101	5	00001110	14
00000110	6	00001111	15
00000111	7	00010000	16
00001000	8		

The last four bits of 16 are the same as the last four bits of 0. If the byte is divided in half each half can be represented as a number between 0 and 15. So for instance

10010110 is (9) and (6) or 96.

This is cumbersome if the value of either half is greater than 9, e.g.

11101011 is (14) and (11).

If 14 and 11 could each be represented by a single symbol the byte could be represented by two symbols, one for each half. To do this new symbols need to be invented for the numbers 10, 11, 12, 14 and 15. It is conventional to use A, B, C, D, E and F.

Thus 11101011 which is (14)(11) can be written EB.

The new numbering system which we have just invented to fit our bytes has 16 symbols (0123456789ABCDEF) and is based on the number 16. Compare this with the decimal system with ten symbols (0123456789) based on the number ten and the binary system with two symbols (01) based on the number 2. It is called the **hexadecimal** system and is often shortened to 'hex'.

Converting from hexadecimal to binary is easy. Use Table 6 below to convert each half.

**Table 6**

Hex	Binary	Hex	Binary
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

A5 hex is converted to binary as follows:

A is 1010 and 5 is 0101

A5 is 10100101.

The conversion of decimal to hex and vice-versa is rarely necessary for our purposes. ASCII codes for letters can normally be looked up in hexadecimal directly. A general method is followed through for the number 213 for completeness.

Step 1. Divide by 16 ignoring any remainder.  
 $213/16=13.3125$  - taken as 13. This is converted directly into the first hexadecimal digit. 13 is D in hex.

Step 2. D0 (hex) would represent  $13 \times 16=208$ . Subtract this from 213 to get the second hex digit.  
 $213-208=5$

Thus 213 is D5(hex).

[On the BBC microcomputer the built in functions `~` and `&` can be used instead. To convert decimal to hex type

`PRINT ~195`

the BBC microcomputer will respond `C3` – the

hex value of 195 decimal.

To convert hex to decimal type

`PRINT &D8`

the BBC microcomputer will respond `216` – the decimal equivalent of `D8` hex.]

## 5 Encoding messages in binary and bar code form

If you do not understand the binary system of numbering a brief explanation can be found in Appendix 4.

The encoding of letters is carried out in four stages.

1. The information to be encoded is first reduced to numbers which should be between 0 and 255. (It would be possible to use bigger numbers but more than 8 bars would be needed. Most microcomputers are at present 8-bit byte machines and numbers over 255 require more than one byte – see Appendix 4).
2. The code number is changed into binary. This means it now appears as a list of eight 0's or 1's.
3. The binary code is translated into bar code. A 0 is represented by a narrow black bar and a 1 by a broad black bar. The broad bars are approximately 4 times as wide as the narrow bars. The black bars are separated by narrow white bars, though the exact width of these is not critical.
4. Normally, to aid the computer in deciding whether a particular code is readable, a 'checksum' is added at the end of each line of bars.

Following this process through for the letters 'MEP'.

1. The letters are coded in ASCII (pronounced asky) form. You should find ASCII tables in the *User Guide* for your computer if you want to refer to them.

letter	—	ASCII code
M	—	77
E	—	69
P	—	80

2. Binary codes are calculated. (A method is suggested in Appendix 4), e.g.

$$77 = 0 \times 128 + 1 \times 64 + 0 \times 32 + 0 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$$

decimal	—	binary
77	—	01001101
69	—	01000101
80	—	01010000

3. The black bar patterns for the individual letters are now worked out. 'n' stands for a narrow bar, 'b' for a broad bar.

		binary	—	bar code
M	—	77	—	01001101 — nbnnbbnb
E	—	69	—	01000101 — nbnnnbnb
P	—	80	—	01010000 — nbnnnnnn

4. The checksum digit is calculated, translated into binary and turned into bars. The checksum is found by adding together the codes for all the letters on the line, and calculating (using the computer if you like)  $sum \text{ MOD } 256$ .

Here total sum of letters is

$$77 + 69 + 80 = 226$$

$$226 \text{ MOD } 256 = 226$$

The checksum is now turned into bar code by stages 1, 2 and 3.

checksum	—	binary	—	bar code
226	—	11100010	—	bbbnnnbn

Note: If the sum is greater than 256 the checksum is always less than 256, e.g.

if sum of digits = 517 then

$$checksum = 517 \text{ MOD } 256 = 517 - 512 = 5$$

The final line of bar code representing 'MEP' plus its checksum is shown in Figure A5.1

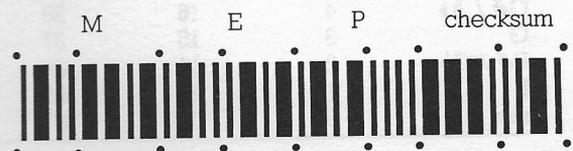


Fig. A5.1 MEP in binary code with checksum

# 6 The music coding system

If you do not understand musical notation then this appendix will not be able to teach it to you and the explanations below will be fairly meaningless. An understanding of bits and bytes is also necessary – see Appendix 4.

Each note has two properties – duration and pitch. These are coded separately, and then combined together to make a single byte. The method is as follows:

- Duration** – Table 7 converts note lengths into numbers. These are subsequently converted into the BASIC sound duration parameters.

**Table 7**

Note value	Code
semiquaver	0
dotted semiquaver	1
quaver	2
dotted quaver	3
crotchet	4
dotted crotchet	5
minim	6
dotted minim	7

Notes shorter than a semiquaver cannot be dealt with. Notes longer than a dotted minim can only be dealt with by repetition – e.g. a semibreve would have to be two minims. No variations in tempo are possible.

- Pitch** – Table 8 converts named notes into numbers. They are subsequently decoded by the program and converted into the BASIC sound pitch parameters.

**Table 8**

Note name	Low octave	Middle octave	Top octave
E	12	24	
D# / E♭	11	23	
D	10	22	
C# / D♭	9	21	
C	8	20	
B	7	19	31
A# / B♭	6	18	30
A	5	17	29
G# / A♭	4	16	28
G	3	15	27
F# / G♭	2	14	26
F	1	13	25
rest	0	0	0

Notes outside the stated range *cannot* be played. Middle C is the low octave C.

These values are combined to form a single byte. The pitch value is a number between 0 and 31 and can be represented by the bottom 5 bits of a byte. This leaves three bits to represent the note length. This is obtained from the table by multiplying the code by 32, e.g. a crotchet A# in the middle octave has the values:

pitch – 18;  
length – 4;  
byte =  $18 + 32 \times 4 = 18 + 128 = 146$ .

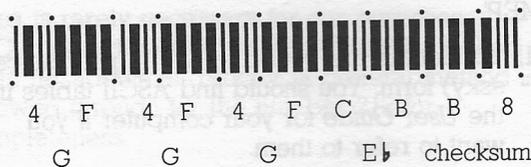
These numbers can be converted into binary (and/or hex) as indicated in Appendix 4 and drawn using the stencil. You will have to calculate and add the check digit for each line. The check digit is calculated as indicated in Appendix 5 on binary messages.

As an example the phrase below is encoded.



- G quaver:G quaver:G quaver:Eb minim  
 $2 \times 32 + 15 = 79$ : (three times):  $6 \times 32 + 11 = 203$
- The bytes are 79:79:79:203
- The check digit is found by adding up  
 $79 + 79 + 79 + 203 = 440$
- Subtract the highest possible multiple of 256  
 $440 - 256 = 184$
- The line of bar code will represent the numbers  
79:79:79:203:184
- In hex these are  
4F:4F:4F:CB:B8

Using the stencil they become

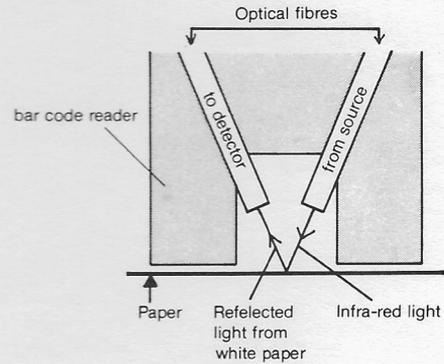


[Answers to the Musical Jigsaw – Worksheet 5.  
Tune 1: Greensleeves – ABACDBDC  
Tune 2: Vicar of Bray – ABAB CDAB CDAB]

## 7 How the bar code reader works

Full engineering details are the subject of a patent and cannot be included in this appendix. The principles, however, are simple.

1. A special transistor emits infra-red light. This light is conducted down an 'optical fibre' (a very fine strand of glass) into a small chamber just above the paper surface. This is the source.
2. White paper reflects the light from the source, while black paper absorbs it. A second optical fibre is sealed into the chamber and any reflected light passes up this fibre. The chamber is illustrated in Figure A7.1.
3. The other end of the second fibre is connected to a light sensitive device which emits a small current if light falls on it.
4. This small current is amplified and is converted into an output voltage which passes along one wire of the cable to the User Port.
5. The programs written now sample the voltage on this wire. While there is a voltage



**Fig. A7.1** How the bar code reader works

light is being reflected which corresponds to white paper. No voltage indicates a black bar.

6. These signals are read as 0 or 1 by the computer and the length of time the signal remains constant can be used to determine the width of the black or white bar.

## THE BAR CODE READER PROJECT

This User Manual provides an introduction to the bar code reader and its uses, with technical details and fault-finding help. Complete coverage is given of all the demonstration programs, together with reproducible black-line masters of worksheets for students, bar code listings and other background information.

© 1984 Council for Educational Technology.  
The materials in this pack have been developed by the  
Microelectronics Education Programme.



**Microelectronics Education Programme**



**Addison-Wesley Publishers**