



## Issue 10 March'83

### IN THIS ISSUE:

- Tape recorder talk
- Flowcharting
- Computer programme review II
- Micronet 800

..... and lots more!

My editorial seems to be getting shorter and shorter every month. Until it is empty completely there's no need to worry though.

Of interest to you this month is the Cabel Electronic Monitors, a 3 day computer workshop, a delay for Proton Acceleration, a toolkit for FORTH and a wordprocessing program. A few LASERBUG matters also included like membership cards, Micronet and a request from you to tell us about the service you get from other people.

Firstly the monitors. The standard monitor for use with the BBC Micro is the Microvitec which although is excellent is a little too much at £300 for a lot of people. The basic monitor available from Cabel Electronics is £199.50 + VAT. It has a full 14" tube (a mullard AX37-590x tube) and is manufactured entirely in Britain. This has a standard RGB socket complete with lead. A Model B is available with a composite video input as well and a Model C with an audio input as well as the other two. If £300 is too much for you then £230 for the Cabel monitor might provide a happy alternative. More details from Cabel Electronics, LLOYD Bank Chambers, The High Street, Tewkesbury, Gloucestershire.

A BBC Micro workshop is to be held at Knuston Hall in Wellingborough, Northants, between the 1st and 3rd of July. The course can be residential if you don't want to keep travelling and costs £27.50 (non-residential price on application). The tutor will be John F Claydon who is known for the North London users group (see meeting place). More details in a future issue of LASERBUG.

The magazine Proton Acceleration has had somewhat of a delay for no real apparent reason apart from a change of address. We'll let you know when something is produced.

Following on from their successful FORTH program for the BBC Micro, Level 9 Computing have launched a toolkit for their FORTH. This enables the program to have a full 6502 assembler, turtle graphics, use double numbers, enables cassette fill handling as well as printer and joystick operation. The toolkit is supplied in separate modules so that you only need to load in what you actually need. It costs £10.00 and is available from Level 9 Computing, 229 Hugenden Road, High Wycombe, Bucks., HP13 5PG. We will be reviewing both the FORTH and toolkit soon in the magazine.

Finally another cassette based wordprocessor is on the market. It is designed for use with the Epson MX80 Type III printer and its list of commands looks impressive apart from the lack of right justification. Versions are available for all different OS and the full features of the Epson printer can be used. The package costs £20.00 and more details are available from Ian Copestake, 23 Connaught Crescent, Brookwood, Woking, Surrey, GU24 0AN. We will hopefully (i.e. if we get a review copy) be looking at the package in softreview.

Oh yes, one more thing before I go into LASERBUG matters. If you are unhappy with tape and can't afford disks there is now an option open to you known as a floppy tape. Basically this is a very fast (and reliable) cassette tape which because of its speed is the next best thing to disks. Its speed is 7500 baud with 101k per cassette possible. Access time averages 22 seconds and up to 138 files can be put on cassette. The system can support two drives and costs £120.00 plus VAT. Details from Ikon Computer Products, Kiln Lake, Laugharne, Carmarthen, Dyfed, SA33 4QE.

Finally a few matters about LASERBUG. Firstly last month you should have all received your membership cards. If you did not then please get in touch. Don't worry if your membership number seems a bit strange as we have completely re-organised the way our files are ordered. If you were our first member you are just as likely to have a membership card number 14 000 whereas if you joined last month you might have card number 1002. Don't worry though as you are on our records no matter what your number.

As we said elsewhere we are now on Micronet 800. You will have to look around to find us as I don't even know where you can see us - when I wrote this editorial it was still 2 weeks before Micronet was due to be launched. Look out though as I am sure you'll be happy with what we provide.

Some of the 1 year members will be getting renewal notices next month. Please don't let your subscription lapse. If you stop receiving magazines now it's not that something's gone wrong with LASERBUG but that your subscription has run out!

Please address all correspondence to:

LASERBUG,  
10 Dawley Ride,  
Colnbrook,  
Slough,  
Berks.,  
SL3 0QH.

Please put one or two words on the top left hand corner of your envelope describing the contents of your letter. We regret that no letter will be replied to unless you enclose a SAE.

Paul Barbour

This is for people with televisions, **not** monitors. As an aside, I cannot understand professional programmers and software houses who produce games or more serious packages entirely on monitors, apparently without ever trying them out on an ordinary TV - in spite of the fact that most BBC Micro users will use a domestic colour TV receiver to connect to their micros. I use a TV, precisely because I want to be sure my programs look good on a TV and not just beautiful on a monitor. (In the same way programs that were written on a TV because of the poorer colours look awful on a monitor, you should try to write your programs so that they look good on both a monitor and TV - Ed.)

So how do you get good colour mixing on a TV without oceans of flicker, orange stipes and waves of purple that chase each other across the screen? As anyone who has tried it will well know, overranging GCOL (for instance using GCOL56,3) usually produces stipey, flickering patterns with a lot of "colour wash" due to the phase of the colour carrier on the PAL system wandering with respect to the main carrier.

I find the best method is as follows. First turn off the interlace using \*TV255,1 (or \*TVsomething,1 depending on where you need to position the picture vertically). Then use alternating colours on odd and even horizontal lines of the picture, that is one colour at Y=0, 8, 16 . . . and different one at Y=4, 12, 20 . . . In practice most of the things one needs to use colour mixing for are constructed from user defined character sets. In this case simply try to use fairly big blocks of colour with a horizontal rather than vertical spread to them. Take a look at the display of a good game like Planetoid or Arcadians and you will see what I mean.

You will find several improvements from this approach. The colours do not "wash" vertically, and although with the interlace off the individual horizontal lines are clearly visible on most TV's, the result is very little flicker and literally a couple of dozen distinct colours, which is all you are likely to need. With the interlace off, completely different effects are produced by putting a colour on the "even" lines Y=0, 8, 16 and by putting it on Y=4, 12, 20 (the "odd" lines). For characters, use:

VDU23,224,0,255,0,255,0,255,0,255

VDU23,225,255,0,255,0,255,0,255,0

(use -1 instead of 255 if you want to save your typing finger - Ed.) and try printing these over all the screen in green, say. The two different greens are quite different (and despite Peter's earlier comments the effect can be seen on a monitor - Ed.).

One further comment, for the more serious reader. If you write programs to sell, you probably want to turn the interlace off without disturbing the vertical sync position which needs to be adjusted by the user according to his/her own TV and micro. So we cannot use \*TV since \*TV,1 is not allowed. Directly addressing the 6845 register 8 using VDU23 is no good either, since the offset set up by any previous use of \*TV is added to any byte you attempt to send to the 6845, with potentially disastrous results.

The answer of course is to discover where in memory the interlace offset is stored. It must be somewhere since it is used at every mode change to switch the interlace on or off. The location concerned turns out to be &24C, and it should be programmed like the second argument of \*TV or \*FX144, that is, to turn interlace off at the next mode change and all subsequent ones set ?&24C=1. This does not disturb the existing vertical sync offset, which is stored at &24B. (All this is very interesting but why on earth not use \*TVQ,1 instead, never use a poke unless you have to - Ed.)

Peter Voke

## contents

● Editorial	2
● Rainbowspot	2
● Assembler Programming on the BBC micro Part III	3
● Oddspot	3
● Flowcharting	4
● Tapereorder talk	5
● Queryspot	6
● Dealer check	6
● Instant memory scan	6
● Business coding	7
● Meeting place	7
● Softreview	8
● Softspot - drive	10
● Softspot - sound-FX	11
● In control	14
● Letters	15
● Software protection part III	15
● Arcade game hi-scores	16
● Useful procedures	16
● Tip time - GCOL	16
● Softspot - tennis/walltrap	17
● Pie chart	18
● Cursor fun	19
● Tip time - *FX in programs	19
● Follow up - conversions for RGB inputs	20
● Treasure hunt - clues about the castle of riddles	21
● Classified	21
● Computer programme review II	22
● Micronet 800	22
● Competition	22
● Computer conversions part IV	22

After my first two articles I hope that people will have some idea of what machine code programming is about. I hope to go on from there to encourage you to sit down with your micro and actually try it out.

Consider the following BASIC routine:

```
L.
 1 A%=65:REM 65 is ASCII code for "A"
 2 X%=40:REM Print "A" 40 times
10 FORI%=1TOX%
20 PRINTCHR$(A%);
30 NEXTI%
```

Remembering that the value of X% is passed to the 6502 X register and that of A% to the accumulator it should be quite easy to write a machine code subroutine to have the same effect.

First re-write the program:

```
>L.
 1 A%=65:REM 65 is ASCII code for "A"
 2 X%=40:REM Print "A" 40 times
10 VDU A%
20 X%=X%-1
30 IF X% <> 0 THEN 10
```

Line 30 is now a conditional branch back to line 10. Now try this, the direct equivalent in machine code:

```
>L.
 1 A%=65:REM 65 is ASCII code for "A"
 2 X%=40:REM Print "A" 40 times
10 DIM P% 50
20 [.PCHR
30 JSR &FFEE
40 DEX
50 BNE PCHR
60 RTS
70 J
80 CALL PCHR
```

```
>RUN
0E8D .PCHR
0E8D 20 EE FF JSR &FFEE
0E90 CA DEX
0E91 D0 FA BNE PCHR
0E93 60 RTS
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA>
```

Now run the program and see that it works. Try altering the values of A% and X%.

Running the program sets up an area of RAM pointed to by PCHR. The first statement (line 30) when executed results in writing the contents of the accumulator to the screen by jumping to an operating system subroutine, OSWRCH. The next operation, DEX, causes the X register to be decremented. In this case X is being used as a count of the characters still to be printed. BNE is short for "Branch if Not Equal to zero" and is the equivalent of the THEN part of the IF statement; it therefore has to follow on from a statement that does the testing part of the IF. In this case it is DEX which does this, so the branch occurs if the result of decrementing X was not 0 i.e. if there are more characters to be printed. If this is not the case then that statement is ignored and the program passes on to RTS and control is returned to BASIC.

Try this example

```
>L.
10 osrdch=&FFE0
20 oswrch=&FFEE
30 DIM P% 50
40 [.PSTR
50 JSR osrdch \ Read Character
60 JSR oswrch \ Write it to the screen
70 CMP #13 \ Carriage return ?
80 BNEPSTR \ Re-do if not
90 RTS \ Return to BASIC
```

```
100 J
110 CALLPSTR
>REM £ means hash (shift-3)
>RUN
0EFF .PSTR
0EFF 20 E0 FF JSR osrdch \ Read Character
0F02 20 EE FF JSR oswrch \ Write it to the screen
0F05 C9 0D CMP #13 \ Carriage return ?
0F07 D0 F6 BNEPSTR \ Re-do if not
0F09 60 RTS \ Return to BASIC
THIS WORKS
```

You will notice that it is commented by the use of the back-slash followed by my comment.

In this program characters are entered one at a time until the RETURN key is pressed. This produces an ASCII code of 13 and hence the use of CMP#13 (CoMPare the contents of the accumulator with the number 13) to test for it. If the character entered was not a carriage return the program branches back to the beginning. Only when CHR\$13 is reached does the compare instruction find it not to be not equal and allow the flow of control to continue to the RTS.

Other branch instructions are available. These are:

- BEQ Branch if equal (to zero)
- BCC Branch if carry clear
- BCS Branch if carry set
- BPL Branch on plus (or zero)
- BMI Branch on minus
- BVC Branch on overflow clear
- BVS Branch on overflow set

In each case a branch is performed depending on a certain aspect of the result of the previous operations. This is possible because most operations set flags within the processor that are tested by the branch instructions. These flags are often called condition codes and are stored in a condition code or status register. Exactly how each instruction effects these flags is best learnt from a book on the 6502 (or a future article from you??? - Ed.) but as an example DEX sets the zero flag if X has been reduced to 0 as a result of the decrementing and sets the sign flag if X has become negative. If the instruction does not set these flags it clears them; on the other hand the other flags are not affected. This is why you need some sort of reference sheet showing which instructions set which flags.

Once you know this you will have much greater control over your problems so it is an important concept, allowing you to create all the structures that are found in BBC BASIC.

Nick Goodwin

## oddsport

This month's Oddsport by the LASERBUG team is an ambiguous mix of MOVE, DRAW and PLOT 85 statements. See if you can guess what it will do before you run it.

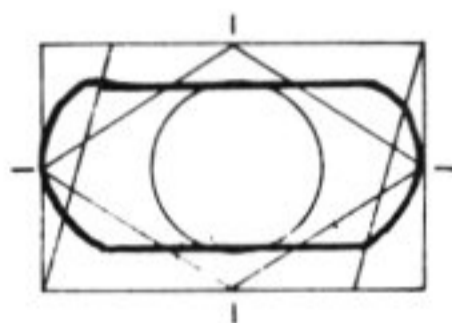
```
L.
 10 MODE2:VDU23:8202:0:0:0;
 20 REPEATVDU29,RND(1280);RND(10
24);
 30 REPEAT:A=RND(7):B=RND(7)
 40 C=RND(7):UNTIL A<>B AND B<>C A
NDR<>C
 50 GCOL0,A:MOVE0,0:DRAW100,0
 60 PLOT85,100,100:DRAW0,100
 70 PLOT85,0,0:GCOL0,B
 80 MOVE100,0:DRAW150,50
 90 PLOT85,150,150:DRAW100,100
100 PLOT85,100,0:GCOL0,C:MOVE0,1
00
110 DRAW50,150:PLOT85,150,150
120 DRAW100,100:PLOT85,0,100:UNT
ILO
```

# 4 flowcharting

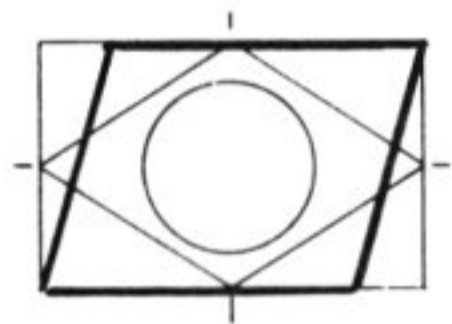


The best way to start off an article such as this is to describe exactly what we mean by the term flowcharting. "A flowchart is a graphical representation of the operations involved in a data processing system. Symbols are used to represent particular operations or data, and flow lines indicate the sequence of operations or the flow of data. A program flowchart is a flowchart used to describe the sequence of operations within a computer program, and forms part of the permanent record of a finished program for maintenance purposes." So there you have it . . . or do you???

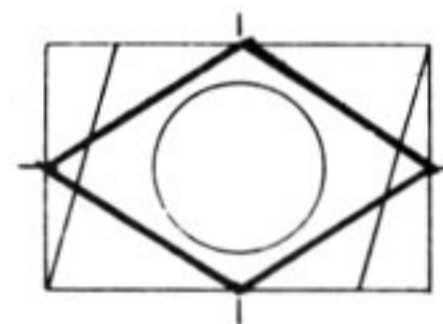
Basically a flowchart is a rough plan of what the program will do where each instruction or part of a program is replaced by a shape. There are many different shapes in flowcharting but for the purposes of this article we will just concentrate on the main ones. The first flowchart symbol is the start/stop symbol: -



You should have one of these at the start of every program you do and another at the end. If you are doing any form of input or output the symbol is a kind of parallelogram:



We will call this the I/O box. One of the main things you do in a program is processing some kind of information. This action is represented simply by a rectangle or suitable size. One more symbol for now. This is one of the most important capabilities of the computer - decision making. The decision box is a triangle with one exit if the case inside the box occurred and another if it didn't:



With these four symbols we can draw up the flowchart of a short program. First the program:

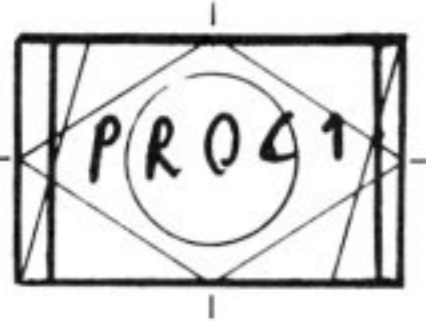
```
>L.
10 INPUT number1
20 INPUT number2
30 number3=number1-number2
40 IF number3<0 THEN 70
50 PRINT "POSITIVE"
60 GOTO 80
70 PRINT "NEGATIVE"
80 END
```

and then the flowchart:



<b>r q FORTH BBC FORTH TOOLKIT</b>			
<p>"r q FORTH" runs on 16K or 32K BBC micros and costs £15. It:</p> <ul style="list-style-type: none"> <li>* follows the FORTH-79 STANDARD and has fig-FORTH facilities;</li> <li>* provides 260 FORTH words;</li> <li>* is infinitely extensible;</li> <li>* has a full-screen editor;</li> <li>* allows full use of the M.O.S;</li> <li>* permits use of all graphic modes, even 0-2 (just!);</li> <li>* provides recursion easily;</li> <li>* runs faster than BBC BASIC;</li> <li>* needs no added hardware;</li> <li>* includes a 70 page technical manual and a summary card;</li> <li>* has hundreds of users.</li> </ul>	<p>Level 9 Computing are pleased to announce a new toolkit for "r q FORTH" on 32K BBC micros. It costs only £10 and adds the following facilities to FORTH:</p> <ul style="list-style-type: none"> <li>* a 6502 assembler, providing machine-code within FORTH;</li> <li>* turtle graphics, giving you easy-to-use colour graphics;</li> <li>* decompiler routines, allowing the versatile examination of your compiled FORTH programs;</li> <li>* the full double-number set;</li> <li>* an example FORTH program and demonstrations of graphics;</li> <li>* other useful routines.</li> </ul>		
<table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> <b>nascom</b>                  Extension Basic . £15/£30 ROM                  Adds 30 new keywords to BASIC                  Compression Assembler 2 . £12                  Small source + high speed             </td> <td style="width: 50%; vertical-align: top;">                 Asteroids ..... m/c,g £7.90                  Galaxy Invaders . m/c,g £5.90                  Missile Defence . m/c,g £7.90                  Super Gulp ..... eb,g £4.90                  5-games cassette . misc £5.90                  (FULL RANGE IN CATALOGUE)             </td> </tr> </table>		<b>nascom</b> Extension Basic . £15/£30 ROM Adds 30 new keywords to BASIC Compression Assembler 2 . £12 Small source + high speed	Asteroids ..... m/c,g £7.90 Galaxy Invaders . m/c,g £5.90 Missile Defence . m/c,g £7.90 Super Gulp ..... eb,g £4.90 5-games cassette . misc £5.90 (FULL RANGE IN CATALOGUE)
<b>nascom</b> Extension Basic . £15/£30 ROM Adds 30 new keywords to BASIC Compression Assembler 2 . £12 Small source + high speed	Asteroids ..... m/c,g £7.90 Galaxy Invaders . m/c,g £5.90 Missile Defence . m/c,g £7.90 Super Gulp ..... eb,g £4.90 5-games cassette . misc £5.90 (FULL RANGE IN CATALOGUE)		
<b>adventures</b> <b>Spectrum BBC nascom</b>			
<p>1) COLOSSAL ADVENTURE: The classic mainframe game "Adventure" with all the original treasures &amp; creatures + 70 extra rooms.</p> <p>2) ADVENTURE QUEST: Through forest, desert, mountains, caves, water, fire, moorland and swamp on an epic quest vs Tyranny.</p> <p>3) DUNGEON ADVENTURE: The vast dungeons of the Demon Lord have survived His fall. Can you get to their treasures first?</p> <p>Every Level 9 adventure has over 200 individually described locations and is packed with puzzles - a game can easily take months to complete. Only sophisticated compression techniques can squeeze so much in! Each game needs 32K and costs £9.90</p>			
<p>ALL PRICES INCLUDE P&amp;P AND VAT - THERE ARE NO EXTRAS. Please send order or SAE for catalogue, describing your micro, to:</p> <p style="text-align: center;"><b>LEVEL 9 COMPUTING</b></p> <p>Dept L, 229 Hughenden Road, High Wycombe, Bucks. HP13 5PG</p>			

If you compare the two you should be able to see clearly the idea of flowcharts. Another symbol you will find essential is the one for PROCedures:



This is like a process box but has two more lines at the edges. By combining these symbols you can form many structures you are used to. For instance instead of FORcounter=1TO10:NEXT you would use:



You should notice that wherever possible in the flowchart you should avoid the use BASIC words. Also don't try to be too detailed in your flowchart – a process box with "calculate quadratic" is much simpler than a box three times the size giving the correct equation. You can always add notes if need be.

It is a very good idea to draw a flowchart for a program **BEFORE** you sit down at your computer and try to write it. This leads to a much better program and in most kinds of examinations in programming is part of the requirements.

Paul Barbour

## tape recorder talk

This is the kind of article that I personally would have thought should have appeared in the first few issues of LASERBUG. However as I still get a couple of letters every week about various aspects of tape recorders I thought an article going into depth about them would still be appreciated – better late than never.

There is one important thing you should remember about your tape recorder – you have paid £300/400 for your computer **BUT** without a tape recorder it is next to useless. Hence it is worth paying out an extra £30 for a new one. Without a tape recorder you can't record your own programs **and** cannot buy software from other companies. Even if you get disks a tape recorder is still essential as at present no company offers software on disk – you still have to load it in from cassette and then save it onto disk (but this will probably be breaking copyright although some companies do allow you to make 2 back-up copies for your own purposes).

The general comment you are given when you ask someone about what tape recorder they should get for their computer is that anyone will do – use one that you have lying around the house. Now this is true but only within reason. A normal domestic tape recorder that you might already have will be fine **if** it has been looked after.

The first and most crucial thing about the tape recorder is the heads which pick up the sound. When these heads are aligned perfectly they can pick up absolutely everything that was recorded on the cassette and when using a music tape this means that you get a very clear sound. However after a while the heads might become disaligned slightly which means that they do not pick up absolutely everything. On a music tape this just means that the sound isn't as clear as it could be. However when using the cassette recorder for your computer this information that the head isn't picking up is absolutely vital to load in the program correctly. Hence as far as your computer is concerned the tape recorder is useless! Now you can take your tape recorder to be serviced and its heads re-aligned but the chances are that the service

agent will tell you that the heads are worn out and need replacing. Now this might well be true but on the otherhand it might not and you might get stung with a bill you shouldn't of had. It might be just as easy to go out and buy a new tape recorder.

Don't go straight out though because there are other possible causes of malfunction. Whilst you are playing tapes through the recorder small amount of the material the tapes are made of (I won't bore you with their scientific name) accumulate on the heads and thus blocking the full signal. If you think that this might be the problem I suggest you purchase a commercial tape head cleaning fluid and a tape head cleaning cassette. Wipe the heads with the fluid *sparingly* and dry them with the brush that should be supplied. Then use the cassette head cleaning tape to finish them off. This might well do the trick.

OK, so now we have (hopefully) a working cassette recorder suitable for your computer – next comes the task of connecting it to your computer. There are three sets of sockets you need to look for:

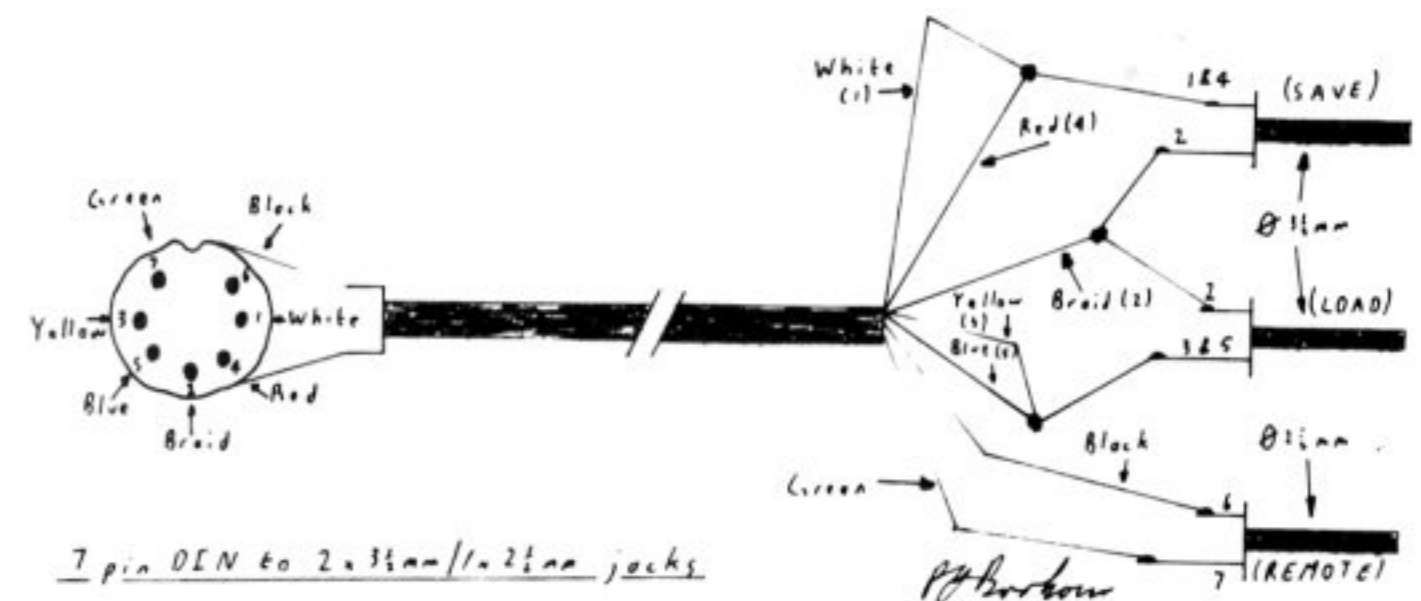
- (i) 5 pin DIN (like the cassette recorder socket on your computer except that it has 2 less pins)
  - (ii) Remote socket (a jack pin socket 2½mm in diameter which may be marked Remote, Rem, I/O, etc.)
  - (iii) A record and playback socket (two jack pin sockets both 3½mm in diameter. The playback socket might be marked with Ear, a picture of an ear, external speaker, etc. The record socket will probably be marked Mic.)
- Which socket (or sockets) you have will determine what lead you require. It is quite normal to have all the sockets – others might just have the 5 pin DIN and yet others might have the jack pin sockets only.

If you have a remote socket (the 2½mm one) then you will have the facility of what is known as motor control. This means that your computer can control whether the tape recorder will work or not. To explain this more simply when you switch on your computer and connect your lead the tape recorder will remain inoperative. if you enter \*MOTOR1 then the cassette recorder LED will light and you are able to operate the cassette recorder properly. When you enter \*MOTOR0 it will be switched off again. Whenever you LOAD/SAVE, etc. a program the light comes on and you can operate your cassette recorder. As soon as the operation is finished the light goes off as does the tape recorder. This feature is extremely handy to have and is invaluable for data file handling (and seeing as the remote socket on my recorder is broken I am feeling the inconvenience of not having the function!).

If on your tape recorder you just have a 5 pin DIN then you really don't need a special lead. All you have to do is go into your local hi-fi shop and purchase a 5 pin DIN to five pin DIN lead. Don't worry that your computer has got seven pins – the extra two pins are to do with motor control. Now the chances are that you also have an earphone socket. You would be well advised either to place an earphone or dummy plug into this socket as normally a five pin DIN does not turn off the main speaker and you would hear the noise of the program loading and saving which is not very pleasant. This should leave you with a perfect system except for the fact that you do not have motor control.

Incidentally although most people supply 5 pin DIN leads in actual fact only three pins are really needed so don't be too worried if you get a 3 pin DIN instead of 5 pins.

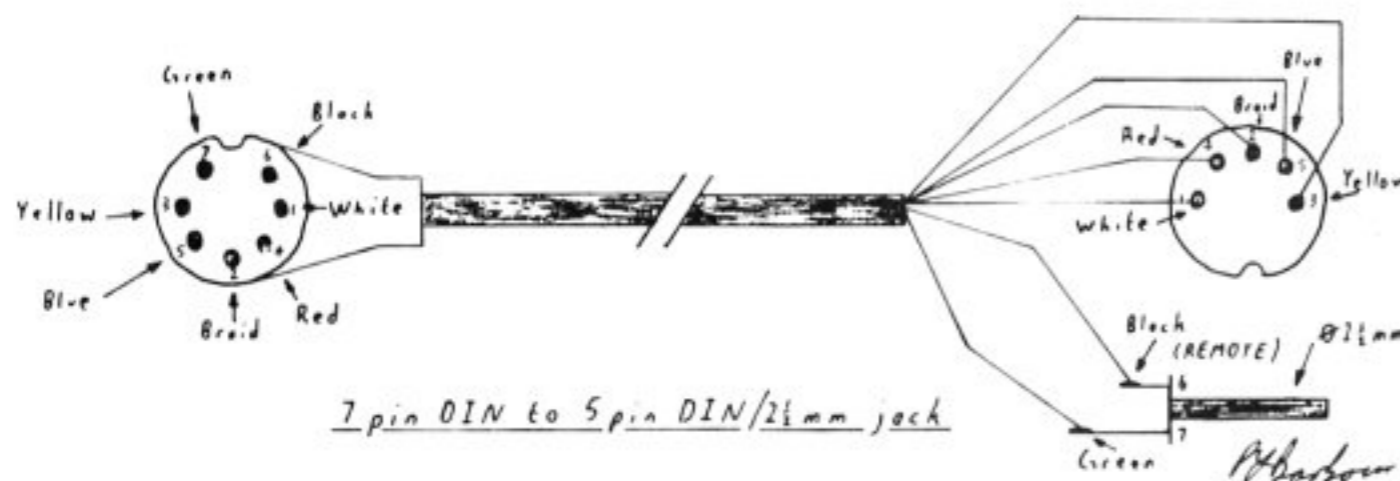
OK, now the next alternative is if you have all 3 jack sockets which I think is the most commonly used format. If you plan to buy a ready made lead you will need to order from an Acorn dealer who sells leads a 7 pin DIN to 2 x 3½mm/1 x 2½mm jacks. Despite what you might of read the average hi-fi dealer does not sell this lead although some specialist electrical/electronics shop (I am sure you know the kind of shop I mean) will sell one over the counter. Now if you want to make the lead up yourself as well as the blank lead from your computer you'll need the 3 jack plugs. Once you have then they should be wired up as outlined in the drawing below:



A quick half hour with LASERBUG and a soldering iron should give you a satisfactory result.

Now supposing you have all of the sockets – which one should you use??? Well, because there are certain differences between different tape recorders the DIN socket is generally accepted to give the best results (although once

you have a working jack plug lead this will also be fine – it's just that the DIN socket has a better chance of a first time success without having to modify the lead. It is however fairly rare that a jack plug lead has to be altered.) The best thing to do is to use the DIN socket for loading and saving and the remote for motor control. For this again you need the blank lead and the appropriate plugs. Armed with your trusty LASERBUG and hot soldering iron (not forgetting a little bit of patience) wire up the lead according to the following diagram:



Now with some tape recorders this isn't the end of all your problems, mainly if you are using the jack sockets. The BBC Micro gives out a very strong signal and on some tape recorders (notably the Ferguson 3TO7 which is the official tape recorder!!!) this signal overloads the tape recorder which means you are not able to SAVE anything yourself. Now before you tell Acorn what they can do with their useless cassette recorder a simple modification to your lead will solve your problem. As the tape recorder is getting too strong a signal what you need to do is weaken this signal by wiring in a resistor in series with the SAVE lead. Now the exact value of this resistor will vary but in the 3TO7 I used to use a 3.9 kilo ohm resistor did the job perfectly. The only reason I chose this value was that the 3.9 resistor was the nearest to hand. Any resistor around this area should work fine.

Just because you have got the entire system set up and the Welcome Tape to load don't think that this is the end of all your troubles. The Welcome Tape is an exceptional program and you will find very few programs recorded as well as this one. As a general rule of thumb have the tone permanently set as high as possible. If you can control whether it is more base or treble have it as high in the treble area as possible. Now the volume setting will determine in general whether or not a program will load. Normally set it to a notch or two below the middle value which should cater with most programs. If you get a program which is awkward to load in then you need to start playing around with the volume. When you alter the volume move it by only half the length of each notch as certain tapes can be this sensitive. If you keep getting Data? or File? errors then you have probably got the volume too quiet. If Header? is the message then it is too loud.

There is one other problem you may find with tapes which I personally have the most trouble with Acornsoft tapes (I first noticed it when I got my copy of Defender). When you first try to load in the program you get nowhere no matter what you do to the volume control. When you rewind the tape and start again you still get nowhere fast. After a few goes the program settles down and loads perfectly. This is a problem more commonly encountered with brand new video tapes particularly of the V2000 format (of which I was unfortunate enough to own for a month or so). The problem is that the tension in the tape is too much i.e. it is wound too tight. After you have played it a couple of times the tension loosens slightly so it can load correctly.

Well, hopefully if you have been having trouble with your tape recorder or if you have just bought your BBC Micro and want to know how to get the CFS set up then you have found something of use in the above article. Do let us know of your experiences with your tape recorder/pre-recorded tapes and particularly if you had a problem and solved it because others might be in the same predicament. If you find any one companies programs very hard to load drop us a line and we'll see what we can do.

Paul Barbour

## dealer check

This is a new section where we want to hear from you about the service your Acorn dealer gives. If you order something from your local dealer keep a record of when you ordered the goods, when they arrived and anything else that is of interest. Let us know if your Acorn dealer gives you an excellent service and we'll praise him. Let us also know though if you get a bad or unsatisfactory service and we'll see what he has to say. Don't limit this to Acorn dealers either – whether its software houses or whatever let us know.

The idea is that if the company you write to us about is good then we can say so in LASERBUG and give them their credit. If the service is bad then not only will we tell other people, we will try and find out what happened and if possible rectify anything you are not happy about.

Just let us know . . .

Dear LASERBUG,

I recently obtained a few copies of your newsletter. The first of these contained this wordprocessor program. Unfortunately the listing has an error, in that no function key is assigned to produce code 135. This has the effect that text cannot be printed. Apart from this one error I found the program to be quite useful. However, this means that there is only one function key left for other purposes, and that was used in the following issue. Anyway, thanks for some good ideas.

R. Henderson, Market Drayton, Shropshire.

**Reply:** *Despite what you think we did set up a key to produce code 135. Enter \*FX4,1 so that the cursor keys return ASCII values. Then try A=GET, press the COPY key and PRINTA – you will find that the copy key returns ASCII code 135. Pressing this key twice causes the screen to be dumped to printer.*

Dear LASERBUG,

On my old ZX81 it was possible to save variables with the program and after loading these remained as long as you used GOTO instead of RUN.

If it is possible to do this on the BBC Micro would you consider advising members how this can be done by way of LASERBUG.

D. Clarke, Woolpit, Suffolk.

**Reply:** *In a word NO, it is not possible to save the variables with your program. You could save the integer variables A% to Z% on tape by \*SAVEing the memory from &400 onwards which is where these variables are stored but apart from that there isn't much I can offer you – unless any member knows different.*

Dear LASERBUG,

I would like to know how to move multi-coloured user defined shapes over other defined shapes without wiping them out and changing the colours as they move.

I have seen this work on Acornsoft's Monster Game.

Mr. D. Tarcton, Woodberry Down, London.

**Reply:** *A good example of what is really happening in this situation is the Acornsoft Arcadians. The way you create this effect is simply to constantly updating the screen so that as soon as you move your shape over the first one and wipe out a part of it you reprint the first shape so that you cannot see the gap. In a program such as Arcadians you cannot re-print the shape fast enough so that for a second on the screen certain shapes are only half there. The listing below gives an example of this happening:*

```
L.
10 MODE1
20 VDU23;B202;0;0;0;
30 FORX%=0TO33
40 COLOUR3
50 PRINTTAB(16,15);"SHAPE 1"
60 COLOUR2
70 PRINTTAB(X%,15)"SHAPE";TAB(X%+6,15);"2"
80 COLOUR0
90 TIME=0:REPEATUNTILTIME=25
100 PRINTTAB(X%,15);SPC(7)
110 NEXT
120 GOTO30
```

Dear LASERBUG,

I am relatively new to computing so have much to learn in all directions. I wonder if there is a routine for 'dumping' pages of teletext (written by your excellent editor for teletext October '82, page 13) into an existing program containing normal line N° listings. There seems little point in saving teletext onto tape and only being able to reload into memory at &7000 or indeed at any address **unless** it's part of an overall idea in a program. Maybe when I start on machine code I will do this myself but in the meantime can you help??

A. Noble, Appleby-Magnam, Staffordshire.

**Reply:** *The answer is simple – just design the pages with our editor and save them onto tape. Then in your program whatever it is just incorporate PROCSCREEN\_LOAD into your program (lines 510-540).*

## instant memory scan

Ever been in the position where you badly need to know what is going on in the memory somewhere around &2DF0, you have a (possibly corrupted) BASIC program lying around in the way, and your tape recorder is playing

up? If this sounds like a particularly revolting adventure game then LASERBUG has the answer.

The easiest way I know of scanning through memory is a one line program available on a function key. The particular one I use looks like this:

```
*KEY8F.A%=B%TOB%+15:C%=?A%.P.A%,C%;:IFC%<127A.C%>31V
DU9,C%.P.:N.:EL.P.:N.|MB%=B%+16^2M
```

The abbreviations used are used to save space in the function key block at &B00 to &BFF, since the definitions are not tokenised. Once programmed, key 8 can be used to step through memory 16 bytes at a time, printing out the location addresses, the contents as a byte in hex, and the translation into a character if there is one. Of course many things that are not characters get translated, but at least variable names, procedures and comments show up clearly.

Setting B% at the beginning sets the starting point for the scan. I also use \*KEY7B%=B%-16|M so that I can move backwards or stand still while altering a byte.

The real advantage of putting a facility like this onto a function key is that it can be loaded into the computer at any time without disturbing anything except the function key block at &B00. Once you have finished the function keys programmed exactly the way you want with RUN, LIST, \*TV or whatever else you want in addition to the scanner you just type \*SAVE"keys1"0B00+0100. This saves the definitions direct onto tape from where they can be loaded direct by the line \*LOAD"keys1". Finally, some interesting locations to start your scan:

```
&800 The envelopes
&C00 Character definitions
&8060 Basic keyword token table
&F150 MOS copyright message.
```

Peter Voke

## businesspot – coding

The idea of this article isn't to help spys with their work but to all you businessmen out there trying to write stock control programs and the like. The article is also useful if you are trying to write a database program for your hobby, etc.

One of the best ways to do such a program is to design a code to suit your needs. All your items can be labelled with this code and makes data interrogation much, much easier. To show you what I mean the easiest way is to give you an example worked through from beginning to end.

For the example I will use one of my own hobbies – Bonsai (for the less educated people among you that is the art of growing miniature trees). The first thing you have to determine is exactly what information you want the code to contain. For the example I want the code to contain (i) the number of the tree, (ii) whether the tree is deciduous or evergreen, (iii) the month and year of the day the tree was started and (iv) whether the date is approximate or exact. The number in your case might be an item number e.g. all boxes of listing paper have the code C1001 for consumables 1 and say printer ribbons C1002 for consumables 2. Alternatively for the hobbist like me they might want each item to have a different number. The other codes really depend on what you require – whether the type of music on a record is Pop or Classical, the sell by date on your product, the colour of the item, etc.

Next you have to determine exactly how big each "field" is to be. For our example I want the number to be one thousand and something preceded by the letter B so that say B1017 is the 17th Bonsai – a field of 5. Whether the tree is deciduous or evergreen can be represented by a single letter, either D or E. The date can be a four digit number so that March '83 becomes 0383 and December '59 becomes 1259. Whether the date is an exact one or just an approximate estimate of the trees life can be shown by E for exact date or A for approximate. Combining these things leaves you with a code as follows: B1aaabccdde

where aaa represents the number of the tree, b what type it is, cc the month of it was started, dd the year and e showing the accuracy of the date. To show this in operation supposing we have an evergreen bonsai started in October 1982 exactly and it is the 21st one in the collection – this gives it a code number of:

```
B1021E1082E
```

which if you check with what we said earlier is correct.

Now not only is this much more convenient it actually cuts down on program variable space as a single string contains 4 pieces of information. If you wanted to list out all the Bonsais that were ten years old you would simply have to use MID\$ to look at the year and print out all entries whose number in this area is 73. This method can be expanded to cover a very wide range of programs. If you are writing a program of the kind talked about earlier seriously consider the idea of coding.

Paul Barbour

And so to our bi-monthly listing of local user groups. First though a few words about the whole subject of local user groups.

When LASERBUG was first set up back in March '82 one of the aims was for us to set up local user groups. However when we first started LASERBUG actually stood for London And South East Region BBCmicrocomputer User Group. Setting up user groups in this area would have been a task we could have coped with. However since those early days LASERBUG has mushroomed at a tremendous rate so that not only did we have to go national, due to the demand we had to turn international. At the last count we had members in 17 different countries. Organising local meetings on a national scale is simply something we are not capable of doing.

Hence we leave the local user group side up to you. If you have already set up a user group then please drop us a line giving full details and we will add you to our user group list – you don't have to belong to LASERBUG to do this.

If you are thinking of starting up your own local user group then two options are open to you. The first is to write to us and get your name put on the Contacts page (which is run on opposite months to Meeting Place). The alternative and probably more effective method is to drop us a line giving details of when and where you are thinking of holding meetings and a phone number that people can get in touch with. Try starting off in your own home for a few months. When you get enough people starting to attend you can think of moving the meetings to a church hall or local school, etc..

Local user group meetings is the other way that members of LASERBUG can get in touch with each other apart from the newsletter. We all at LASERBUG feel that this is a very good way of getting more out of your computer. You can share your experiences with others and get the benefit of their discoveries.

If there is no user group near you we would strongly advise you if you can spare an evening or two a month to seriously consider starting up your own group. It only needs one person to make the first move.

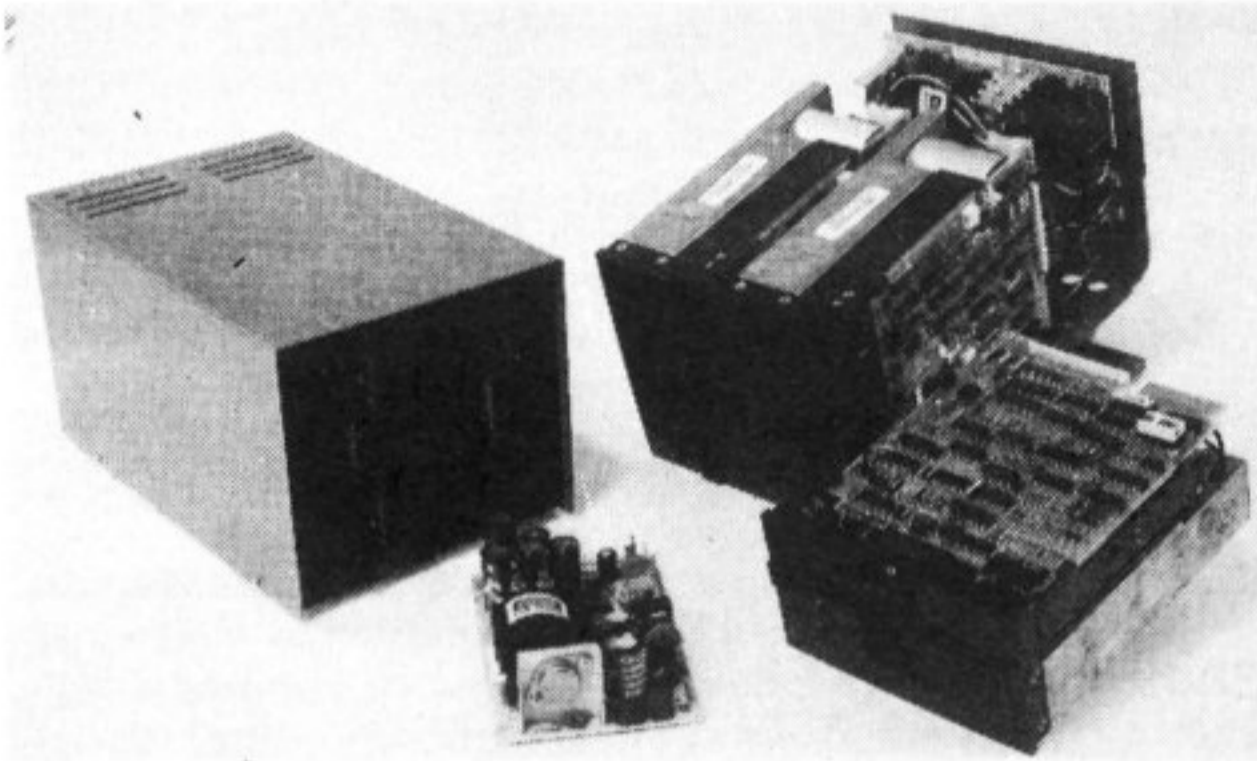
We do offer an affiliation to local user groups as long as one person in the group is a member of LASERBUG (preferably the leader of the group). If your user group would like to be affiliated to LASERBUG then please drop us a line. Affiliation to us does give you several advantages. For a start this provides you with a "big brother" with whom you can refer to for anything. Secondly there is our Club Reports page which is reserved for anything the affiliated clubs want to use it for whether it be the news of a big event or just a report on your last meeting. Thirdly we offer club discounts on subscriptions – if three or more people from one individual club wish to subscribe to LASERBUG as long as all the copies of the magazine are sent to the main address and distributed from there we are prepared to offer a 10% discount on each individual fee. Apart from that if affiliated clubs need any other help that we can offer let us know and we will see what we can do.

Anyway, the present list of local user groups is:

- |                  |  |
|------------------|--|
| Geoff Barker,    | Cardiff BBC Computer Club (CBCC). Penarth 701023. Holds meetings on alternate Wednesday evenings in the Applied Science Lecture Theatre of University College, Newport Road, Cardiff. Extensive facilities at the Lecture Theatre. After only three months had 60 members. |
| James Bridson,   | Barnsley Computer Users Group, 39 Keresforth Hall Road, Kingstone, Barnsley, S. Yorkshire, S70 6NF. 0226-41753 (after 4.30 p.m. please).   |
| John Claydon,    | North London BBC Microcomputer Users Group and Education Workshop. 01-889 5446. Meetings held at Bounds Green Junior School, Park Road, N11 on the second Sunday of each month at 2.00. Fee of approx. £1/meeting to cover costs. AFFILIATED TO LASERBUG.                  |
| G.W. Goodacre,   | CHELMERBUG, 34 Quilp Drive, Chelmsford, CM1 4YA. Formal meetings held on the first Wednesday of each month at a local school. Informal meetings in-between in members homes.   |
| Nick Goodwin,    | South-West Wales BBC and Electron User Group (allied with the Swansea Computer Club at present). Meetings every Tuesday above the Three Lamps pub in Swansea. AFFILIATED TO LASERBUG.  |
| Nick Lamb,       | 23 Gaywood Close, Caister-on-Sea, Great Yarmouth, Norfolk, NR30 5RD. 0493-728442.  |
| Norman Lambert   | Orpington Computer Club, 11 Vinson Close, Orpington, Kent, BR6 0EQ. Meetings held every Friday evening at a local church hall.   |
| Janne Soderberg, | Frihetsvagen 32, S-175 33 Jarfalla, Sweden. Phone 0758-317 53 (caters for the Atom as well as the BBC Micro).  |

All enquiries about Meeting Place or Affiliation should have the words User Group in the top left-hand corner of the envelope.

## MICROWARE ANNOUNCE THE Z/ L RANGE OF FLOPPY DISC SUB SYSTEMS



The Control Data Family of Flexible Disk Drive is a single- or double-sided, random-access, low-cost storage device. Maximum storage capacity 1 megabyte on a 133.4-millimetre (5.25-inch) interchangeable disk. This disk drive is interchangeable with comparable products, providing industry compatibility in size, mechanical mountings, electrical interface, power requirements and physical appearance. CDC® FDD operates in single- or double-density formats. Single-density operation is achieved by using Frequency Modulation (FM) encoding, and in double-density operation uses Modified Frequency Modulation (MFM).

### OPERATIONAL CHARACTERISTICS

	9408	9409	9409T
Capacity			
Unformatted			
Per disk	250.0 kbytes	500.0 kbytes	1 megabyte
Per track	3.1 kbytes	6.2 kbytes	6.2 kbytes
Formatted (16 sectors, 126/256 bytes)			
Per disk	163.84 kbytes	327.68 kbytes	655.36 kbytes
Per track	2.1 kbytes	4.1 kbytes	4.1 kbytes
Code	MFM	MFM	MFM
Transfer Rate	125 kbits/s	250 kbits/s	250 kbits/s
Average latency	less than 100 ms	less than 100 ms	less than 100 ms
Seek Time			
Track to track	less than 5 ms	less than 5 ms	less than 5 ms
Average Access	less than 80 ms	less than 80 ms	less than 132 ms
Setting time	less than 15 ms	less than 15 ms	less than 15 ms
Head Load Time (OPT)	less than 50 ms	less than 50 ms	less than 50 ms
Media	hard/soft sector	hard/soft sector	hard/soft sector
Rotational Speed	300 r/min	300 r/min	
Track Density	48 TPI	96 TPI	
Flux Reversal Density			
(track 39, side 1)	5876 FRI	5922 FRI	
Number of Tracks	40	80	
Inner recorded radius (side 0)	1.437 in (36.50 mm)	1.385 in (35.2 mm)	
Outer recorded radius (side 0)	2.250 in (57.2 mm)	2.250 in (57.2 mm)	
Inner recorded radius (side 1)	1.354 in (39.39 mm)	1.344 in (34.1 mm)	
Outer recorded radius (side 1)	2.167 in (55.0 mm)	2.167 in (55.0 mm)	

### THE Z/ L RANGE

Type	Capacity	Tracks	No. of Drives
ZL141	250 k	40	1
ZL142	500 k	40	2
ZL241	500 K	80	1
ZL242	1 Mb	80	2
ZL291	1 Mb	160	1
ZL292	2 Mb	160	2

Microware Disk Drive Subsystems are plug compatible with the following:—

TANDY, VIDEO GENIE, BBC MODEL B, NABCOM, SUPER BRAIN, IBM/PC. and many more.

Microware also supply floppy drives, Winchesters and sub systems to OEM's and the trade at very aggressive prices, offering the best price performance ratio available. Call and ask for details.

### CABINETS AND PSU

Cabinets of steel, aluminium or plastic are available in a choice of colours, and each unit is fully guaranteed.

### AC 8151 40 WATT SERIES

The AC 8151 Switch-Mode Power Supply has been designed for use in small terminals and other similar equipment. The AC 8151 has dual line inputs and regulated outputs of: +5V at 2.5A, +12V at 2.0A, -12V at 0.1A. This compact unit features low magnetic radiation and is built to conform to International Safety and RFI Regulations.



Microware (London) Ltd.,  
637a Holloway Road, London N19 5SS.  
Telephone 272 6398/6237

**EDUCATIONAL PROGRAM:** Educational 1

**REQUIREMENTS:** 32k

**SUPPLIER:** Golem, 77 Qualitas, Bracknell, Berks., RG12 4QG

**PRICE:** £8.05 (+ VAT + 50p P & P)

**DESCRIPTION OF PROGRAMS:** This tape contains 7 different programs suitable according to the maker for ages 5 to 9. The first program on the tape is an addition or subtraction test. The computer and the child both have a ball which is moved along a line as the child gets a question right. If he gets it wrong then the computer's ball moves forward—the first one to reach the end of the line is the winner. Full error trapping is included so that the child cannot press the wrong key. Pressing ESCAPE takes you back a step to the menu, etc. The next program is the same as the first but involves multiplication. Cubecount follows on—the computer draws a shape using cubes and you have to say how many cubes make up the picture. Sometimes this can involve pure guesswork and I can't really see how this program is educational??? A shape recognition program follows which asks the child to identify 3 shapes out of 8 possible ones. This program is fine although there is no check to see that the child isn't asked the same question twice in a row. Next is a memory testing game—triangles, squares and circles are shown on the screen in different colours. The screen is then cleared and the child is asked a question about what was on the screen. An attempt at a spelling test is the penultimate program. First you have a choice of how many words you want to guess and then their standard. You can see the words for a short time and then have to re-enter them. Finally you have a clock tester program which takes so long to actually draw the clock face you get very bored! The actual program is OK but like a lot of these clock tester programs still has the emphasis on entering the time in numbers not in words. This is essential if the child is ever to learn properly and can be done as we did an issue or two ago. Overall a reasonable selection of educational programs.

**PRESENTATION:** ★★★

**FOR AGES:** 5 – 9

**SUBJECTS:** 3 maths/1 english/3 misc.

**USEFULNESS:** ★★★

**NUMBER OF USERS:** 1

**LOADING PROBLEMS:** Yes (had to use a 2nd tape recorder)

**VALUE FOR MONEY:** ★★★

—o0o—

**UTILITY PROGRAM:** Disassembler

**REQUIREMENTS:** 16k (a version is supplied for both 16k and 32k)

**SUPPLIER:** David Knell, 13 Northumberland Road, Leamington Spa, Warwickshire, CV32 6HE

**PRICE:** £5.00

**DESCRIPTION OF PROGRAM:** This tape is proof that the good software doesn't always come from the big companies. The tape contains a very comprehensive disassembler, monitor and much more. It is recorded at 300 baud which is very unusual and once entered operates from 22 different commands. On top of this all OS commands (i.e. those preceded by an \*) can be used. The best way to show the power of this program is to tell you about its commands—output memory in ASCII, create backup copy of the program, disassemble continuously, dump memory, execute from location, fill set area of memory with specified code, output memory in hex, disassemble, find all occurrences of specified code within set area, enter monitor, scroll forwards or backwards, move memory from one area to another, put string into memory, change location, print string at specified location . . . The list just goes on. This program is very good and at £5.00 is excellent value for money.

**PRESENTATION:** ★★★

**RESPONSE SPEED:** ★★★★★

**SIZE:** 2.37k

**USEFULNESS:** ★★★★★

**LOADING PROBLEMS:** No (because recorded at 300 baud)

**VALUE FOR MONEY:** ★★★★★

—o0o—

**MISCELLANEOUS PROGRAM:** The Computer Programme Programs II

**REQUIREMENTS:** 32k

**SUPPLIER:** BBC Soft, 35 Marylebone High Street, London, W1M 4AA

**PRICE:** £10.00

**DESCRIPTION OF PROGRAMS:** Similar to the volume 1 reviewed a few issues ago but the programs are longer and can only be run with 32k. Eight programs make up this package. The first using the method of turning on and off colours produces a spinning globe. Next is the famous fish pond program solving that victorian puzzle—firstly with a graph and secondly by showing the event graphically. An excellent demonstration of the difference between parallel and serial data transfer follows which really shows up the point quite clearly. The middle program on the tape gives a break from the more serious programs and plays a reasonably good lunar lander. Well worth a look at by itself. Next is an interesting example of how an LED works. Two more



programs left. The first is a geography quiz program which you can alter to ask your own questions. The last one is a noughts and crosses game in which the computer starts off by knowing nothing about the game, learns by its mistakes and finally plays a very, very good game. Unlike volume 1 this tape does contain some useful rather than interesting programs. Again like the first one I thoroughly enjoyed reviewing this package.

**PRESENTATION:** ★★★  
**USE OF GRAPHICS:** ★★★★★  
**LOADING PROBLEMS:** No  
**VALUE FOR MONEY:** ★★★

—o0o—

**BUSINESS GAME PROGRAM:** GB Ltd.  
**REQUIREMENTS:** 32k  
**SUPPLIER:** Simon W Hessel, 15 Lytham Court, Cardwell Crescent, Sunninghill, Berks.

**PRICE:** £5.95  
**DESCRIPTION OF PROGRAM:** Fancy taking your chances at running the country??? If so then this program might interest you. The idea is for you to take on the role of Prime Minister and decide how the financial affairs of the country is to be run. If you manage to stay in office long enough for a general election then you have to take your chances along with the other parties. If you are into this sought of program then GB Ltd. will certainly prove a good buy. A different type of game to the normal "zap" types and very interesting. Worth taking a look at.

**PRESENTATION:** ★★★★★  
**STANDARD OF GAME PLAYED:** ★★★★★  
**LOADING PROBLEMS:** No  
**VALUE FOR MONEY:** ★★★★★

—o0o—

**ADVENTURE PROGRAM:** Arrow of Death  
**REQUIREMENTS:** 16k  
**SUPPLIER:** Digital Fantasia, 24 Norbreck Road, Norbreck, Blackpool  
**PRICE:** £8.95

**DESCRIPTION OF PROGRAM:** This is a different type of adventure program from the standard one, in scenario at least. However not wanting to spoil your fun if you buy this program I won't say any more. Unlike most of the adventures it is suitable for 16k and makes the effort to use colour in MODE7. A good adventure and one worth adding to your collection.

**PRESENTATION:** ★★  
**COMPLEXITY:** ★★★★★  
**RESPONSE SPEED:** ★★★★★  
**LOADING PROBLEMS:** No  
**VALUE FOR MONEY:** ★★★

—o0o—

**EDUCATIONAL PROGRAM:** Flags  
**REQUIREMENTS:** 32k  
**SUPPLIER:** IJK Software, 9 King Street, Blackpool, Lancs.  
**PRICE:** £4.50

**DESCRIPTION OF PROGRAM:** This program gives you a number of flags from different countries. With each flag you must say what country it comes from, the capital of that country and the continent. Although you are given a list of all possible answers when the program was tried out on children as an example the American Flag was shown. The child entered "USA" and the computer said that it was wrong and should have been "U.S.A.". He then entered the capital as "WASHINGTON D.C." and was told again that he was wrong and the correct answer was "WASHINGTON"!!! You can't win!!! **IF** you follow the program and use only the answers given then the program might be useful for a geography test?

**PRESENTATION:** ★★★  
**FOR AGES:** 14+ (some of the questions were very hard)  
**SUBJECT:** Geography (Flags)  
**USEFULNESS:** ★★★  
**NUMBER OF USERS:** 1  
**LOADING PROBLEMS:** No  
**VALUE FOR MONEY:** ★★★

—o0o—

**MISCELLANEOUS PROGRAM:** Predict  
**REQUIREMENTS:** 16k  
**SUPPLIER:** Mayday Software, 181 Portland Crescent, Stanmore, Middx., HA7 1LR

**PRICE:** £4.99  
**DESCRIPTION OF PROGRAM:** This program is a competent pools predictor which if you think means it guesses the depth of swimming pools or something means I had better explain a little further. The idea is that you enter into the program data about how the football teams are playing and the computer will predict how you should fill in your football pools coupon,

hoping that you should win a fortune! The program comes complete with an 8 page manual telling you how to work it and was very nicely presented. If you want your chance on the pools to be a little less chancey then try this program.

**PRESENTATION:** ★★★  
**LOADING PROBLEMS:** No  
**VALUE FOR MONEY:** ★★★

—o0o—

**PROGRAM PACK:** BBC Octuplet  
**REQUIREMENTS:** 32k  
**SUPPLIER:** R.W. Darlington, 1 Fells Grove, Worsley, Manchester, M28 5JN

**PRICE:** £4.95  
**DESCRIPTION OF PROGRAMS:** If I had the opportunity to offer a software pack of the year award then this program would win it beating Acornsoft and whoever else produces software for the BBC Micro. I think that this pack offers any BBC Micro user the best value for money that he will ever see in a program pack and would recommend all of you to purchase this pack. The pack contains 8 different programs, several of which would be worth at least £4.95 if they were sold by themselves. A full list of instructions is supplied. The first program is called Digitiser which is a very comprehensive graphics program comparable with those costing £15+!!! It allows fairly complex pictures to be drawn including magnification and shrinking of the picture drawn. The second program can be tagged on to the end of your own programs to allow pictures produced by the first programs to be used in your own programs. Thirdly is Dominoes which again is worth £5 by itself. This is a very good game and makes interesting playing. Fourth is Strip Jack Naked, cross between strip poker and snap! Don't worry though, the computer just plays the card part. Next is a program to set up the user definable keys. Three more programs left. Sixth is a slight variation on Battleships. Seventh is Morsecode where you enter your own message and the computer will display it as morse. The last program is one to define your own characters with hundreds of pre-defined symbols afterwards. What more can I say, at £5 you find me a program that is better value for money.

**PRESENTATION:** ★★★  
**USE OF GRAPHICS:** ★★★★★  
**LOADING PROBLEMS:** No  
**VALUE FOR MONEY:** ★★★★★★ (and the normal maximum is five stars!!!)

—o0o—

**ARCADE GAME PROGRAM:** Car Wars/Alien Planet  
**REQUIREMENTS:** 32k  
**SUPPLIER:** Software for All, 72 North Street, Romford, Essex  
**PRICE:** £6.95

**DESCRIPTION OF PROGRAMS:** Two fairly straightforward programs on this tape. The first is Car Wars similar to the program on the BBC Soft Fun Games pack except that I think this version is much, much easier without a shadow of a doubt. The other one is Alien Planet where your spaceship is crashing and you have to drop bombs on the city below to flatten it before you crash into one of the towers. Two standard games for a standard price, written to a good standard.

**PRESENTATION:** ★★★  
**USE OF GRAPHICS:** ★★★  
**ADDICTIVE QUALITY:** ★★  
**LOADING PROBLEMS:** Yes (second tape recorder had to be used)  
**VALUE FOR MONEY:** ★★★

—o0o—

**MISCELLANEOUS PROGRAM:** Cube Master  
**REQUIREMENTS:** 32k  
**SUPPLIER:** Acornsoft, 4a Market Hill, Cambridge, CB2 3NJ  
**PRICE:** £9.95

**DESCRIPTION OF PROGRAM:** The idea of this program is two-fold. Firstly you can solve a Rubiks Cube on the screen or set the screen up to your own Rubiks Cube and let it solve your own own. Unfortunately the program is far from perfect. When I first loaded the program in I asked it to solve an already solved cube – it took 8 moves. I asked it to solve the cube it had just solved and it took 54 moves – to solve a correct cube! The program is OK but not one of Acornsoft's better efforts.

**PRESENTATION:** ★★★  
**USE OF GRAPHICS:** ★★★★★  
**LOADING PROBLEMS:** No  
**VALUE FOR MONEY:** ★★★

LASERBUG would like to thank Golem, David Knell, BBC Soft, Simon W Hessel, Digital Fantasia, IJK Software, Mayday Software, R.W. Darlington, Software for All and Acornsoft for supplying us with review material.

>GOTO1260  
 DRIVE - (c) LASERBUG 1983  
 This is an early arcade game, seen in the fifties and probably long before. Use the joystick to keep on the winding road. It consists of several procedures. init 120-390 provides instructions and builds special characters. start 400-500 allows the road width to be changed on start or restart. track 510-620 reduces the road size on leaving the starting area. run 630-730 decides which way the road should go and uses procs to print two lines causing scrolling. single 740-770 draws the normal road. split 780-820 incorporates an island. joy 830-920 checks the joystick using ADVAL1 for steering and ADVAL2 for speed control -line 910 controls the time in the loop. score 930-1060 makes allowance for speed and road width  
 h  
 Line 90 checks for a crash.

Enhancements could include running the game for a fixed number of steps (SZ) instead of stopping on a crash and adding time penalties when off the road;  
 adding a second player, perhaps on a higher or lower line as a handicap.  
 If you don't want the island obstacle Change 720 to PROCsingle.

```
>L.
10 REM      DRIVE
20 REM      By Alan Stoner
30 REM      and Simon Treen
40 :
50 REM      January'83
60 :
70 REM      Version 1.0
80 :
90 REM      Takes up 4.09k memory
100 :
110 REM     Requires 32k
120 :
130 REM     Written on OS 0.1
140 :
150 REM     (c) LASERBUG 1983
160 :
170      :::::
180 :
190 MODE 7
200 PROCinit
210 PROCstart
220 MODE5: COLOUR0: COLOUR130 : CLS
230 PROCtrack
240 REPEAT
250 PROCrun
```

```
260 PROCjoy
270 UNTIL (POINT(PPZ,511)=2) OR (POINT(PPZ+63,511)=2): REM until crash!
280 PROCscore
290 GOTO210
300 DEFPROCinit
310 PRINT"DRIVE on the road"
320 PRINT""Use right hand joystick"
330 PRINT" with button towards you."
340 PRINT"Start with joystick in the centre"
350 PRINT" and move it sideways to steer,"
360 PRINT" away from you to go faster,"
370 PRINT" and towards you to slow down."
380 PRINT"Fire to restart after a crash."
390 PRINT"Enter width of road 3, 4, 5 or 6"
400 PRINT"Default is 6 which is easiest"
410 PRINT"PRESS THE FIRE BUTTON TO START"
420 VDU 23,255,255,255,255,255,255,255,255,255
430 VDU 23,254,255,255,254,254,252,252,248,248
440 VDU 23,253,0,0,1,1,3,3,7,7
450 VDU 23,252,240,240,224,224,192,192,128,128
460 VDU 23,251,15,15,31,31,63,63,127,127
470 VDU 23,250,128,128,192,192,224,224,240,240
480 VDU 23,249,127,127,63,63,31,31,15,15
490 VDU 23,248,248,248,252,252,254,254,255,255
500 VDU 23,247,7,7,3,3,1,1,0,0
510 VDU 23,246,&E7,&C3,&B1,0,0,0,0,0,0
520 VDU 23,245,0,0,0,0,0,&B1,&C3,&E7
530 A$=CHR$(245)
540 B$=CHR$(246)
550 WZ=6: ZY=0
560 GCOL0,3
570 ENDPROC:REM #####
580 DEFPROCstart
590 REPEAT
600 W=INKEY(10)
610 IF W=51 THEN WZ=3
620 IF W=52 THEN WZ=4
630 IF W=53 THEN WZ=5
640 IF W=54 THEN WZ=6
650 UNTIL ADVAL(0)=1
660 XZ=10-WZ/2 :REM WZ=ROAD WIDTH, XZ=LEFT HAND BORDER
670 GZ=WZ DIV 2
680 ENDPROC:REM #####
690 DEFPROCtrack
700 FOR IZ=1 TO 20
710 PRINT SPC(1);STRING$(18,CHR$(255))
720 NEXT
730 FOR IZ=1 TO 6
740 PRINTSPC(IZ+1);STRING$(18-2*IZ,CHR$(255))
750 PRINTSPC(IZ+1);STRING$(18-2*IZ,CHR$(255))
760 NEXT
770 HZ=(RND(10)*10)
780 SZ=0 :REM £ OF CYCLES ROUND LOOP
790 NOW=TIME :REM used in scoring
800 ENDPROC:REM #####
810 DEFPROCrun
820 AZ=RND(11)-1
830 SX=SZ+1
```

```

840 REM change the border width unless changing dir
ection
850 IF Z% = 0 THEN X% = X% - 1: Z% = 254: GOTO 900
860 IF X% > AZ + 1 GOTO 890
870 IF Z% = 250 THEN X% = X% + 1 ELSE Z% = 250
880 GOTO 900
890 IF Z% = 254 THEN X% = X% - 1 ELSE Z% = 254
900 IF S% = H% PROCsplit ELSE PROCsingle
910 ENDPROC:REM #####
920 DEFPROCsingle
930 PRINT SPC(X%);CHR$(Z%-1);STRING$(W%-1,CHR$(255
));CHR$(Z%)
940 PRINT SPC(X%);CHR$(Z%-3);STRING$(W%-1,CHR$(255))
;CHR$(Z%-2)
950 ENDPROC:REM #####
960 DEF PROCsplit
970 H% = (RND(10)*25)+S%
980 PRINT SPC(X%);CHR$(Z%-1);STRING$(G%-1,CHR$(255)
);B%;STRING$(W%-G%-1,CHR$(255));CHR$(Z%)
990 PRINT SPC(X%);CHR$(Z%-3);STRING$(G%-1,CHR$(255))
;A%;STRING$(W%-G%-1,CHR$(255));CHR$(Z%-2)
1000 ENDPROC:REM #####
1010 DEFPROCjoy
1020 P% = ADVAL1 DIV 52 :REM CONTROL STEERING
1030 VDU 5
1040 MOVE P%,527 :REM STRADDLES CENTRE LINE
1050 PRINT "*"
1060 VDU 4
1070 PP% = B%*(P% DIV 8) :REM calculate pos of *
1080 Q% = ADVAL2 DIV 1536
1090 Z = INKEY(Q%)
1100 ENDPROC:REM #####
1110 DEFPROCscore
1120 SOUND 0,-15,100,20
1130 VDU 28,0,31,19,19
1140 COLOUR 131:COLOUR 0
1150 CLS
1160 PRINTTAB(1,2);"STEPS";TAB(14)S%
1170 SECONDS% = 1+(TIME-NOW)/100
1180 PRINTTAB(1,4);"SECONDS";TAB(14)SECONDS%
1190 SPEED% = S%/SECONDS%
1200 PRINTTAB(1,6);"WIDTH";TAB(14)W%
1210 PRINTTAB(1,8);"SPEED";TAB(14)SPEED%
1220 SCORE% = S%^2/SECONDS%/W%
1230 PRINTTAB(1,10);"SCORE";TAB(14)SCORE%
1240 ENDPROC:REM #####
1250 REM:COMMENTS FOR ARTICLE - NOT MEANT AS PART OF
LISTING
1260 PRINT"DRIVE - (c) LASERBUG 1983"
1270 PRINT"This is an early arcade game, seen in"
1280 PRINT"the fifties and probably long before."
1290 PRINT"Use the joystick to keep on the winding ro
ad."
1300 PRINT"It consists of several procedures."
1310 PRINT"init 120-390 provides instructions and"
1320 PRINT" builds special characters."
1330 PRINT"start 400-500 allows the road width to"
1340 PRINT" be changed on start or restart."
1350 PRINT"track 510-620 reduces the road size on"
1360 PRINT" leaving the starting area."

```

```

1370 PRINT"run 630-730 decides which way the road"
1380 PRINT" should go and uses procs to print"
1390 PRINT" two lines causing scrolling."
1400 PRINT"single 740-770 draws the normal road."
1410 PRINT"split 780-820 incorporates an island."
1420 PRINT"joy 830-920 checks the joystick using"
1430 PRINT" ADVAL1 for steering and ADVAL2 for"
1440 PRINT" speed control -line 910 controls the"
1450 PRINT" time in the loop."
1460 PRINT"score 930-1060 makes allowance for speed a
nd road width"
1470 PRINT"Line 90 checks for a crash."
1480 PRINT"Enhancements could include running"
1490 PRINT"the game for a fixed number of steps"
1500 PRINT"(S%) instead of stopping on a crash"
1510 PRINT"and adding time penalties when off"
1520 PRINT"the road;"
1530 PRINT"adding a second player, perhaps on a "
1540 PRINT"higher or lower line as a handicap."
1550 PRINT"If you don't want the island obstacle"
1560 PRINT"Change 720 to PROCsingle."

```

## softspot – sound-fx

The program below is an editor for sound and envelope commands (although with no graphical display). It is written in MODE1 because I wanted to use colour-coded text, yellow for values, etc. There's so much information on the screen your eyes need a bit of help to sort it out. MODE7 having fewer lines crowds the text up too much. There's no reason why it shouldn't be MODE4 compatible though so it could run on a model A but the colour commands would need sorting out. I can't at a quick glance see any reason why it shouldn't work on a series 1 OS (nobody ever can, until they find the program doesn't work – Ed.).

It's mostly self-explanatory I think – a simple note A is installed on loading (in fl) and all the info in a program can be saved to tape. Parameter C accepts hex values (whether or not you type in the ampersand) – thanks to an earlier edition of LASERBUG. Retaining with D=0 zeroes all parameters on the key specified (remember to use the digit keys when specifying which function key is required). But they can anyway be overwritten. If you hold down a function key the note will keep repeating until it empties the keyboard buffer. I haven't inhibited this because it can at times be a useful facility. The program hangs if you overload any of the sound queues when using 'sync' control on parameter 5 (i.e. by pressing the first half of a dual sound more than four times before pressing the second/third halves – 3 halves? Well, you know that I mean.) ESCAPE clears the queues leaving everything else unchanged. It could be inhibited but I haven't bothered, it's not important.

I hope you all find it useful.

```

L.
1 REM      SOUND-FX
2 REM    by Patrick Dowling
3 :
4 REM      January'83
5 :
6 REM      Version 1.0
7 :
8 REM    Takes up 4.96k memory
9 :
10 REM    Requires 32k
11 :
12 REM    Written on OS 0.1
13 :
14 REM    (c) LASERBUG 1983
15 :
16 :::::
17 :

```

```

18 REM DO NOT RENUMBER PROGRAM !
19 :
20 :::::
21 :
140 MODE1
150 ON ERROR GOTO2370
160 *KEY0 "19 :M"
170 *KEY1 "49 :M"
180 *KEY2 "50 :M"
190 *KEY3 "51 :M"
200 *KEY4 "52 :M"
210 *KEY5 "53 :M"
220 *KEY6 "54 :M"
230 *KEY7 "55 :M"
240 *KEY8 "56 :M"
250 *KEY9 "57 :M"
260
270 PROCinit
280 VDU19,0,4;0;19,3,1;0;
290 REPEAT PROCmenu:UNTIL menu$="Q"
300 MODE6:END
310 *****
320
330 DEFPROCmenu
340 CLS:VDU19,1,4;0;
350 COLOUR3
360 PRINTTAB(4,3)"*** SOUND & ENVELOPE EDITOR ***"
370 COLOUR1
380 PRINTTAB(0,7)"Go to EDIT mode - - - - - <
E >"
390 PRINT"Load data from tape-file - - < L >"
400 PRINT"Store data on tape-file - - - < S >"
410 PRINT"Print data of sounds retained < P >"
420 PRINT"Replay retained sounds - - - f1 - f9"
430 PRINT"To exit programme - - - - - < Q >"
440 PRINTTAB(0,21)"fkeys used:":FOR k=1 TO 9:PRINT
key$(k);:NEXT
450 PRINT"Set DUR.=0 to cancel fkeys."
460 VDU19,1,6;0;
470
480 REPEAT
490 PRINTTAB(0,19)SPC38"
500 INPUTTAB(33,19),menu$:v=VAL(menu$)
510 IF v<=57 AND v>=49par=v:PROCreplay
520 IF v>0 THEN 490
530
540 IF menu$="P" PROCprint:menu$="M"
550 IF menu$="S" PROCstore:menu$="M"
560 IF menu$="L" PROCload :menu$="M"
570 IF menu$="E" PROCedit
580 IF menu$="Q" ENDPROC
590 UNTIL menu$="M" OR menu$="M"
600 ENDPROC
610
620 DEFPROCedit
630 CLS:VDU19,1,4;0;19,3,4;0;
640 COLOUR1
650 PRINTTAB(8,1)"Env. -- F R E Q U E N C Y --"
660 PRINT TAB(15)"step rate no.of steps"
670 COLOUR3

```

```

680 PRINTTAB(6)"1 2 3 4 5 6 7 8"
690 COLOUR1
700 PRINTTAB(6);"N T P11 P12 P13 PN1 PN2 PN3"
710 REMCOLOUR3
720 PRINT TAB(8,10)"Env. -- A M P L I T U D E ---"
730 COLOUR3
740 PRINT TAB(14) "9 10 11 12 13 14"
750 COLOUR1
760 PRINT TAB(13)"AA AD AS AR ALA ALD"
770 REM C.3
780 PRINT TAB(8,18)"Sound Chan. A/Env. Ptch. Dur."
790 COLOUR3
800 PRINT TAB(15)"C";TAB(21)"A";TAB(28)"P";TAB(34)"D"
810 COLOUR1
820 PRINTTAB(0,27)"Listen<fkey0> Retain<R> Replay<
f1-9>"
830 PRINTTAB(0,29)"fkeys used:"
840 VDU19,1,6;0;19,3,1;0;
850 REPEAT
860 IF A(15)>9 amp$="&" ELSE amp$=" "
870 VDU19,2,4;0;
880 PRINTTAB(11,29)SPC28"
890 COLOUR1:VDU31,11,29:FOR k=1 TO 9:PRINT key$(k)
;:NEXT
900 COLOUR2
910 PRINTTAB(0,7)SPC38"
920 VDU11:PRINT"ENV. ";A(1) " ";A(2)TAB(14)A(3)T
AB(18)A(4)TAB(22)A(5)TAB(27)A(6)TAB(31)A(7)TAB(35)A(8)
930 PRINTTAB(0,15)SPC38"
940 VDU11:PRINTTAB(13);A(9);TAB(17);A(10);TAB(21);
A(11);TAB(25);A(12);TAB(30);A(13);TAB(34);A(14)
950 PRINTTAB(0,21)SPC38"
960 VDU11:PRINT"SOUND"TAB(21);A(16);TAB(28);A(17);
TAB(33);A(18):VDU31,14,21:PRINTamp$;"A(15)
970 IF N>0 AND par>19 COLOUR1:VDU11:PRINTTAB(0,18)
" (f";N)"
980 VDU19,2,3;0;
990
1000 COLOUR3
1010 INPUT TAB(6,24)"Parameter 1-18 (M=menu):",p$:p
ar=VAL(p$)
1020 COLOUR1
1030 IF par=19 PROCplay
1040 IF p$="C" par=15
1050 IF p$="A" par=16
1060 IF p$="P" par=17
1070 IF p$="D" par=18
1080 IF par<58 AND par>48 PROCreplay
1090 IF par<19 AND par >0 PROCrange:COLOUR2:INPUT T
AB(24)"Value:",val$:A(par)=VAL(val$)
1100 COLOUR1:IF par=15 PROCchex
1110 IF p$="R" OR p$="r" PROCretain
1120
1130 PRINTTAB(0,18)SPC6"
1140 PRINTTAB(31,24)SPC7"
1150 PRINTTAB(0,25)SPC38"
1160 UNTIL p$="M" OR p$="m":menu$="M"
1170 ENDPROC
1180
1190 DEFPROCchex

```

```

1200 IFLEFT$(val$,1)<>"&" val$="&"+val$
1210 A(15)=EVAL(val$)
1220 ENDPROC
1230
1240 DEFPROCrange
1250 rng$="Range: "
1260 IF par=16 GOSUB1340:ENDPROC
1270 ON par GOSUB 1350,1360,1370,1370,1370,1380,1380,
1380,1390,1400,1410,1410,1420,1420,1430,0,1440,1460
1280 range$=rng$+r$
1290 PRINT range$;
1300 ENDPROC
1310
1320
1330 REM Subroutines:
1340 PRINT"Amp.-15 to 0/Env.1 to 4";:RETURN
1350 r$="1 to 4":RETURN
1360 r$="0 to 127 (x "+CHR$(224)+CHR$(225)+CHR$(229)+
")":RETURN
1370 r$="-128 to 127":RETURN
1380 r$="0 to 255":RETURN
1390 r$="0 to 127":RETURN
1400 r$=CHR$(228)+" 127":RETURN
1410 r$="-127 to 0":RETURN
1420 r$="0 to 126":RETURN
1430 r$="0 to 3 (or Hex)":RETURN
1440 IF A(15)=0 r$="0 to 7(C=0)":RETURN
1450 IF A(15)<>0 r$="0 to 255":RETURN
1460 r$="-1 to 254 (x "+CHR$(226)+CHR$(227)+)":RETUR
N
1470
1480 REM End of subroutines
1490
1500 DEFPROCplay
1510 ENVELOPE A(1),A(2),A(3),A(4),A(5),A(6),A(7),A(8)
,A(9),A(10),A(11),A(12),A(13),A(14)
1520 SOUND A(15),A(16),A(17),A(18)
1530 ENDPROC
1540
1550 DEFPROCretain
1560 REPEAT:PRINTTAB(0,25)SPC38""
1570 INPUTTAB(12,25)"which number fkey ",N:UNTIL N<
10 AND N>0
1580 par=20
1590 FOR K=1 TO 18
1600 IF A(18)=0 A(K)=0:par=0
1610 M(N,K) = A(K)
1620 NEXT
1630 key$(N)=" f"+STR$(N)
1640 IF M(N,18)=0 key$(N)="":par=0
1650 PROCplay
1660 ENDPROC
1670
1680 DEFPROCreplay
1690 N=VAL(CHR$ par)
1700 FOR K= 1 TO 18
1710 A(K) = M(N,K)
1720 NEXT
1730 PROCplay
1740 key$(N)=" f"+STR$(N)
1750 IF M(N,18)=0 key$(N)="":par=0
1760 ENDPROC
1770
1780 DEFPROCprint
1790 CLS:VDU14
1800 FOR N=1 TO 9
1810 COLOUR3:PRINT"f";N
1820 COLOUR2:PRINT"ENV. ";
1830 FOR K=1 TO 13
1840 PRINT;M(N,K)", ";
1850 NEXT:PRINT;M(N,14)
1860 PRINT"SOUND ";
1870 IF M(N,15)>9 PRINT"&";
1880 PRINT;"M(N,15)", ";
1890 FOR K=16 TO 17
1900 PRINT; M(N,K)", ";
1910 NEXT:PRINT;M(N,18):NEXT
1920 VDU15:G=GET
1930 N=0:ENDPROC
1940
1950 DEFPROCstore
1960 VDU28,0,31,39,19,12,26:REM window
1970 INPUTTAB(0,20)"If data-tape is ready to RECORD,"
" enter file name... "A$
1980 IF A$="" ENDPROC
1990 *OPT1,1
2000 X=OPENOUT(A$)
2010 FOR N=1 TO 9
2020 FOR K=1 TO 18
2030 PRINT EX,M(N,K)
2040 NEXT:NEXT
2050 CLOSE EX
2060 N=0:ENDPROC
2070
2080 DEFPROCload
2090 VDU28,0,31,39,19,12,26:REM window
2100 INPUTTAB(0,20)"If data-tape is ready to PLAY,"""
enter file name... "A$
2110 *OPT1,1
2120 X=OPENIN(A$)
2130 FOR N=1 TO 9
2140 FOR K=1 TO 18
2150 INPUT EX,M(N,K)
2160 NEXT:NEXT
2170 CLOSE EX
2180 N=0:ENDPROC
2190
2200 DEFPROCinit
2210 VDU23,224,65,194,68,73,19,33,65,129
2220 VDU23,225;0;34,85,85,85,34
2230 VDU23,226,65,194,68,78,17,34,68,135
2240 VDU23,227,5,5,0,32,80,80,80,32
2250 VDU23,228,24,24,126,24,24,0,126,0
2260 VDU23,229,80,80,0;0;0;
2270 DIM A(18), M(10,18),key$(9)
2280 menu$="":par=20:N=1
2290 FOR K=1 TO 18
2300 READ A(K):M(1,K)=A(K):NEXT
2310 FOR k=1 TO 9:key$(k)="":NEXT
2320 key$(1)=" f1"

```

We all know that holding the CTRL key down while pressing another key normally produces a special control character, which, when sent to the VDU chip to be printed, causes some special action to be taken. The user guide tells us how to use quite a few of these, but in case any of you are not yet in the habit of using these useful key combinations, here is a list of the most common ones:

CTRL-A send next character only to printer  
 CTRL-B turn printer on  
 CTRL-C turn printer off  
 CTRL-H move cursor (also CTRL-I, J or K)  
 CTRL-L clear text screen; cursor to top left  
 CTRL-M the same as return key  
 CTRL-N scrolling off, paging on  
 CTRL-O paging off, scrolling on  
 CTRL-T set all colours back to normal (default palette)  
 CTRL-P clear graphics screen  
 CTRL-U delete the line being entered  
 CTRL-Z reset both graphics and text windows to whole screen  
 CTRL- cursor to top left

What is not so widely known is that all the other possible control codes can also be issued direct from the keyboard, without using a VDU statement, although in some cases it is more convenient to type in the corresponding VDU statement. What is more, the way these other control codes work can be understood easily since they follow the format of the familiar VDU statements precisely.

The difficulty is that the other control codes expect one or more further bytes of information to be sent following the first one, the subsequent bytes being data. For instance VDU22 is the same as MODE (no, no, no. *If you use VDU 22 instead of MODE you will find that your computer will crash fairly soon, especially if you go into MODEs 0, 1 or 2 in that way with a decent length program in the memory. When you use VDU22 although the screen mode changes, the value of HIMEM is not altered which can lead to a fatal error. The rule is if you use VDU22 (or the control key equivalent) the first command afterwards should reset the value of HIMEM as suitable for the MODE you are in. For the Model B in MODEs 0, 2 and 2 HIMEM=&3000, MODE3 HIMEM=&4000, MODEs 4 and 4 HIMEM=&5800, MODE 6 HIMEM=&6000 and MODE 7 HIMEM=&7C00 - Ed.*) and expects one further byte to specify the mode. So press CTRL-V, CTRL-A and you should get mode 1 - a quicker way of getting there than typing MODE1 once you are used to it (*I would have thought that MODE1 was quicker than CTRL-V, CTRL-A, HIMEM=&3000 - Ed.*) The reason it works is that CTRL-V is actually the code 1 (see table of ASCII codes in the appendix of the User Guide). CTRL-V, CTRL-G will give you MODE7, CTRL-V, CTRL-@ will give MODE0. There is an even easier way; since the computer interprets MODE&31 as mode 1 just press CTRL-V,1 (by this I mean press V with CTRL held down following by 1 without CTRL) to get mode 1. The same principle works for the other modes. However do not get misled, the computer does not simply accept the number 1 as 1; it is taking the ASCII code of 1 which is &31 and reducing it modulo 8 to get 1.

When using other control codes, one has to be more careful since the computer is not so obliging. For instance CTRL-Q is the same as VDU17 which in turn is the same as COLOUR and expects one extra byte to specify the colour. CTRL-Q,CTRL-A should give colour 1 which it does (Try it in modes 2 or 5). CTRL-Q,1 also does the same thing, but if you are in mode 2 you will have some difficulty in selecting colour 10 for your text. What comes after 9 in the ASCII order? I find it easier to do CTRL-Q,CTRL-J if I really want colour 10 - not very likely!

CTRL-R selects graphics colours exactly like GCOL. However CTRL-R,1,2 will **not** be the same as GCOL1,2; instead it will be taken by the computer as GCOL49,50 and give the usual stripey effects that result from overranging GCOL. To switch to colour 2 in graphics, use CTRL-R,CTRL-@,CTRL-B.

CTRL-S is more useful in practice. If you like to list in mode 6 with a blue background, you can only do it via a VDU19 statement normally. Try CTRL-S, CTRL-@,CTRL-D,CTRL-@. This should give a blue background if you typed it exactly right. It is exactly equivalent to VDU19,0,4,0,0,0 but in my opinion easier to type once it is familiar.

Other control codes work similarly. The ones that normally require large numbers (graphics coordinates) sent with a semicolon after them have to be done as two bytes and hence are impractical in the extreme. These are VDU24 (CTRL-X) which sets up the graphics window, VDU25 (CTRL-Y) which is the PLOT command and VDU29 (CTRL-]) which moves the graphics origin. VDU23 is not very practical as CTRL-W, since the characters one can redefine have codes 224 to 255 and none of these can be entered from the keyboard. But the experts may like to reprogram the 6845 direct from the keyboard as under VDU23; For a dramatic look at zero page, go into mode 2 and then type CTRL-W,CTRL-@,CTRL-L and then press CTRL-@ seven times.

Text windows can be set up using CTRL- , which is the same as VDU28, provided it is followed by the appropriate four bytes. If you are constantly

using the same text window, this could be handy. It is also possible to move the text cursor around the screen directly using CTRL-£ followed by the text co-ordinates. For instance if you want to start at 5,5 then type CTRL-£, CTRL-E, CTRL-E.

I haven't mentioned CTRL-G since it merely emits an annoying beep, nor CTRL-D and CTRL-E which are familiar VDU4 and VDU5.

They might be useful at the keyboard if you want to type in at two different positions on the screen alternately. I can imagine an interesting text editor working on this principle. Anyone have the time spare? CTRL-[ is the same as ESCAPE.

The only one left is CTRL-@ which does nothing. The only thing to do is to invent a use! There must be better ideas, but I use CTRL-@ as an alternative "edit finish" code when copying large amounts of text using the edit keys. The edit finish code is stored at &37E and it normally contains 13, the ASCII code for RETURN so the RETURN key is what signals the end of editing. To switch this off set ?&37E=9. Editing will then continue until CTRL-@ is pressed. You can press return at the end of each copied line and the edit cursor just stays there, up where you want it.

Peter Voke

## a number at random . . .

When is a random number not a random number? Answer, when it's a pseudo-random number. Second answer, especially if it is a pseudo-random number generated by the BBC micro.

The RND function supplied by BBC Basic is enormously useful in all sorts of situations ranging from games to Monte Carlo integration methods. Whatever your particular obsession, you are sure to use it some time or other. But be careful. The numbers that come out may not be as random as you would like them to be.

For instance try this:

```
>L.
10 ONERRORVDU15:END          80 PRINT
20 K%=1                       90 NEXT
30 @%=&303                    100 END
40 VDU14
50 FORI%=1TO1000
60 A=RND(-I%*K%)
70 FORJ%=1TO8:PRINTRND(8);:NEXT
```

Don't forget the ; after the RND(8), otherwise the pattern is not so obvious.

You see what I mean. The set of numbers is about as random as the sleepers on the London to Birmingham line. The fault of course is in the use of RND(-) which, in technical jargon initialises the random number generator with the number after the minus sign - in the program above the number is I%\*K% (in non-technical jargon it is impossible for a computer to think of random numbers - how on earth could a computer pick the first number that came into its head within certain limits when it hasn't any reasoning powers of that sort. Inside the computer it has a very, very, very long list of numbers which because of the lists length and the distribution of numbers is sufficiently random for any purpose you would have. As it cannot produce truly random numbers these are called pseudo-random numbers. Using RND with a minus argument in actual fact resets whereabouts in this list the computer starts from next. This is only a rough guide for what happens as the very technically minded among you will know more about the random number generator than that - Ed.) The extraordinary thing is that the sequences are so similar even though the value of I% keeps on changing. If I% did not change, we would expect the sequence to be the same, of course. What we discover is that initialising the function with two numbers that are similar produces two very similar sequences. I think of all sorts of useful (or amusing) uses for that knowledge.

To check this idea, replace line 20 in the program to K%=3456789. This, obviously, makes the initialising numbers in RND(-I%\*K%) more random. But be careful that you don't use this in circumstances where I%\*K% could become too large for the computer. K% has to be quite big if the sequences are to lose all trace of pattern. Try replacing line 10 by K%=34567 or indeed set K% to virtually and four or five figure number.

A popular trick is to use the TIME to initialise the random number generator, with a line: A=RND(-TIME). This is fine but still be careful. Change line 60 suitably and try a run. Hardly very promising.

Used with care RND can produce superb patterns, exciting and unpredictable games, as well as having other useful applications. But care is needed to avoid the traps set by random numbers that actually pseudo-random. If you don't watch out, you might end up as I did once with 1500 supposedly random words - all of which began with S!

Peter Voke

Dear LASERBUG,

I note with interest the letter from John Shaw on page 4 of Issue 8 (January '83). The following observations may be relevant or at least of interest:

- (i) The instructions given with the Bugbyte chess game are incorrect (at least with my copy). One should type either \*RUN"BEEBCHESS" or just \*RUN.
- (ii) In common with a disproportionate amount of Bug Byte BBC Software the chess game will not run with the OS 1.0. This has been proven by members of our local group.
- (iii) At least two UK dealers have advised us not to chance Bug Byte software if OS 1.0 is fitted.

Even if the aforementioned points do not solve John Shaw's problem at least he can feel comforted in the fact that he's not alone.

Paul Derbyshire, Netherlands.

Dear LASERBUG,

My sample copy of "LASERBUG" has given pleasure. It looks to be a goodish journal. A pity, therefore, that on page 17 (November '82) you gave way to yucky remarks about "The wife" and "Coronation Street" even if this is offset by "The husband" at his "Space Invaders" rather than his machine coding! But I did think seriously that after Dragon with the ad "Read this to your wife . . ." we'd never see such a thing in a Beeb journal! Seriously though, I think you should do all you can to ignore or if necessary minimise any imagined male/female split in the micro world as this is important.

Mrs. J.M. Tillet, Norwich.



Mr. P. Barbour - Editor,  
Laserbug,  
10 Dawley Ride,  
Colnbrook,  
Slough,  
Berks SL3 0QH.

12th January 1983.

Dear Sir,

Mr H. R. Lees Model B Computer.

Further to Mr Lees letter to queryspot with regard to his model B computer, I would like to express the official (ACORN) definition of a model B computer. The BBC machine is one computer with two possible specifications. The model B must fulfill the B specification, contain ACORN supplied components, and be assembled by an approved agent of ACORN. In our particular case, we are an APPROVED service centre, with a REGISTERED CHARTERED ENGINEER as head of department, STANDARDS that are traceable to the NATIONAL PHYSICAL LABORATORY, and a workshop equipped with a complete set of ACORN'S own computerised test equipment for testing BBC computers.

As regards the cost of the computer, which Mr Lee paid £36.00 (9% above the BBC charge, this was our widely advertised price and was fixed by the well known fact that dealers had to buy the computers directly from the BBC at full price. In our case we added a surcharge of 9%, which was a break-even figure for selling the computer, testing before depatch (1 hr soak test) and providing the customer with full back up. I must point out that due to the non-profit situation, a large number of dealers would not handle the BBC computer, whilst others felt it necessary to add a 15% surcharge.

Finally, what could be of interest to your readers, is that I will be only too pleased to show them around our service centre. If anyone is interested, I would suggest they contact me.

Yours faithfully,

Dr. C. N. Wormald.

c.c. Mr. H. R. Lees, 50 Nicholson Road, Healing, Grimsby, DN37 7NW  
c.c. Mr. S. Dow, Sales Manager, ACORN COMPUTERS LIMITED.

217 Manningham Lane, Bradford, West Yorkshire, BD8 7HH. Tel: Bradford 722512.  
Associated Co. of Eltec Services Ltd. Directors: Dr C N Wormald C Eng MIEE, Catherine E Tweddle BSc.  
Registered in England No. 1524262.

Dear LASERBUG,

You pointed out in the last edition of LASERBUG (Issue 8 - January '83, page 8 - Ed.) that in order to take Acorn to task for failing to upgrade all the BBC Micro's with 0.1 OS one would need to prove that the earlier machines did not match the specification published at the time the computer was

launched. I have studied carefully the literature I received when I first considered purchasing my machine and I must admit that, as far as the detailed technical specification is concerned it is very difficult to show that Acorn has failed to deliver what was promised. The documents in question merely describe the facilities offered - they do not guarantee that they are entirely bug free.

The fact remains, however, that I believed I was buying a machine which would provide "versatile and straightforward file control" whereas I have had to abandon all attempts to load data into my lengthy programs from cassette because of the nasty OS bug which makes it impossible to load reliably from a series of files. Great things were expected from the BBC Micro by educationalists like myself. I spend hours weekly trying to develop modern languages software for schools, most of this time being spent at home on my own machine. It took a long time to solve the problem of French and German accents. The limitation on the redefinition of characters came as a great disappointment. Acorn were quite unable to provide information on this facility for superimposition in the graphics mode, in spite of the fact that it is described in the manual (for which I had to wait nearly a year). The 10k I lose in order to get a few acutes, graves and circumflexes leads to a regular encounter with the deadly "No Room" announcement (language programs consume large quantities of memory) and forces me to struggle in vain to load in my data from cassette files.

All in all my experiences with the 0.1 OS are summed up in the words disappointment and frustration. The latter increased a hundred fold when I consider that the work I am doing, if successful, will help to sell more BBC Micros to Scottish schools!

The story does not end there. At Christmas I purchased from Acorn their LISP and FORTH packages. I received the LISP cassette but no manual and no FORTH. A phone call to Cambridge extracted the information that items were not available. I acquired another version of FORTH from Level 9 Computing (by return of post). My request for a refund from Acorn has not even been acknowledged, nor was receipt of an article which I had produced for our regional modern languages newsletter, describing the application of the BBC Micro to the teaching of my subject. Do they ever answer anyone or acknowledge anything? Why did the BBC not give the microcomputer project to Clive Sinclair?

J. Campbell, Elton.

## software protection pt. III

First, a comment on the making of "backup" copies of commercial machine code programs (please note that when you purchase a program, whether or not you are legally allowed to make backup copies of programs depends on the actual company. Although sometimes you are entitled to make 2 backup copies for your own purposes only, in general even this is breaking copyright - Ed.). Apart from the hint which appeared in LASERBUG a month or two ago (Issue 8, page 2, Letters - Ed.) to use \*OPTI,2 before \*LOADing the program, there is one other tip that should be mentioned: don't check the program is okay after loading by CALLing the execution address (the execution address is the one given last on loading after \*OPT1,2). This is a mistake since some commercial programs rewrite the contents of the execution address and other locations as soon as they are run, and hence cannot be broken out of and rerun. This is a simple self destruct mechanism which ensures that the program in memory after a run is useless. You should \*LOAD the program and then \*SAVE it immediately using the load address, length and execution address given under \*OPT1,2.

On the question of protecting your own programs, there is no way of preventing fresh backup copies being made by the above method with the BBC Micro, so far as I know. The best protection for programs that have real commercial value is to put a copyright message into the program itself, or even print one on the screen. If writing machine code, this defends your copyright against anyone who cannot also understand machine code. To protect yourself against the experts, you will have to make the program depend on subtle and complex ways on the actual contents of the copyright message itself; not an easy task and probably not worth it.

For the majority of us who write in BASIC, the best way of defending an embedded copyright message against prying eyes and fingers is the use of non-printing control characters. Essentially you put a REM into the program which contains a lot of nonsense (a series of 0's or A's for instance) which are then subsequently changed into control characters by byte indirection. The control characters I use are 127 (delete), 21 (turns the screen off altogether), 6 (turns the screen back on), and 2 & 3 (which controls printers) (try also using other combinations of the ASCII control codes - quite startling effects can be produced - Ed.).

Here is how to do it. Consider the line: 100 PRINT"(c) D.Duck 1983":REMAaaaaaaaaaaaaaaaaaaaaaaaaaaaaaREM. You type this line in

and then look through memory until you find the string of a's (we have printed several programs/routines for looking at the memory locations – try looking through Issues 8, 9 and this one, 10 – Ed.). They will appear as a string of 97's since 97 (&61) is the ASCII code for a. Before then you should find the byte &F4 which is the token byte for REM. All you do is change all twenty-six a's into character 127 (delete) by a loop like this:  
FOR I%=&E7B TO &EA1: ? I%=127: N.

This changes all the 97's representing the a's into 127's representing deletes with backspacing. Result? When you, or anyone else lists the program, line 100 looks like this:

```
100REM
```

The REM is the one I put in after the a's, and it is only there to mislead the inquisitive into thinking that line 100 is an ordinary blank REM line. What has happened is that the 26 deletes after the real REM deletes all the 26 characters in front, in other words just back to the beginning of PRINT. In fact there is nothing to stop you deleting back to the beginning of the line number or beyond.

The use of characters 21, 6, 2 and 3 is similar except that after attempting to list character 21 the screen will turn itself off altogether until a subsequent listing if character 6. This can be used to suppress listing of whole subroutines at a time – though mismatching line numbers after a renumber will give their presence away. Characters 3 and 2 are used in a similar way to turn off printers and turn them back on, though less predictable effects on printers with buffers. Of course all these characters must be part of REM statements in the BASIC otherwise they will cause errors to occur.

Peter Voke

## arcade game hi-scores

This is a new section that has been promised for several issues. On this page we will list a number of arcade-type games available for the BBC Micro and the high scores that people have obtained on them. All the scores below are those the LASERBUG team managed to get after a short while with most of the games and all should be very simple to beat. Please go ahead and send us in your top scores but please (i) have a witness to sign the letter and (ii) tell us roughly how long it took you. I'm sure that we will dream up a prize for the best score. Anyway, the high scores are:

Arcadians (1): 7 600  
Atlantis (2): 26 300  
The Frog (3): 15 050  
Galactic Firebird (4): 9400  
Invaders (2): 3 060  
Meteors (1): 9 500  
Monsters (1): 10 860  
Planetoid (1): 44 500  
Stratobomber (2): 5 984  
Zombies (3): 2 870

(KEY TO SUPPLIERS: (1) Acornsoft, (2) IJK Software, (3) Software for All, (4) Kansas – generally all these games are the best of their kind in this topic i.e. we thought that Invaders from IJK was the best invaders program available.)

## useful procedures

This is the first in an occasional series of useful PROCedures. If you are writing a graphics program it is often handy to know the exact X,Y coordinates of a set square. I found this vital in writing One Armed Bandit. So, for anyone who would find the same PROCedure useful it is listed below. Use ZX:/ to move the small cursor and press Y when you want to know the coordinates of the spot.

```
>L.
```

```
10000 DEFPROCEDIT
10010 LOCALX%,YZ,A$
10020 X%=640:YZ=512
10030 REPEATPLOT69,X%,YZ
10040 A$=GET$
10050 PLOT71,X%,YZ
10060 IFA$=":" THENYZ=YZ+4
10070 IFA$="/" THENYZ=YZ-4
10080 IFA$="X" THENXZ=XZ+4
10090 IFA$="Z" THENXZ=XZ-4
```

```
10100 UNTILA$="Y"
10110 PRINT"X=";XZ;" , Y=";YZ
10120 ENDPROC
```

Paul Barbour

## tip time – GCOL

Whenever you set a graphics colour most people just use GCOLOR. Very few ever attempt to use a first number other than 0. In actual fact the second number sets the logic of the line drawn. If all that sounds complicated then do not despair – open the user guide at page 262 and enter the short program below and you can experiment to your hearts content!

```
>L.
10 REM GCOL DEMO
20 REM by Paul Barbour
30 :
40 REM Version 1.0
50 :
60 REM (c) LASERBUG 1983
70 :
80 : : : : :
90 :
100 MODE7:VDU23;8202;0;0;0;
110 PROCdata
120 MODE1:VDU23;8202;0;0;0;
130 PROCdraw
140 GOTO100
150 END
160 :
170 : : : : :
180 :
190 DEFPROCdata
200 PRINT""Enter colour 1 (0-16)"
210 INPUTcol1%
220 IFcol1%<0ORcol1%>15THEN200
230 PRINT""Enter GCOL number 1 (0-3)"
240 INPUTgcol1%
250 IFgcol1%<0ORgcol1%>3THEN230
260 PRINT""Enter colour 2 (0-16)"
270 INPUTcol2%
280 PRINT""Enter GCOL number 2 (0-3)"
290 INPUTgcol2%
300 IFgcol2%<0ORgcol2%>3THEN280
310 IFcol2%<0ORcol2%>15THEN260
320 PRINT""Enter background colour (0-16)"
330 INPUTbgrnd%
340 IFbgrnd%<0ORbgrnd%>15THEN320
350 ENDPROC
360 :
370 DEFPROCdraw
380 VDU19,1,col1%,0,0,0
390 VDU19,2,col2%,0,0,0
400 VDU19,0,bgrnd%,0,0,0
410 GCOLgcol1%,1
420 FORX%=590TO690STEP4
430 MOVEXZ,0:DRAWXZ,1024
440 NEXT
450 GCOLgcol2%,2
460 FORYZ=462TO562STEP4
470 MOVE0,YZ:DRAW1280,YZ
480 NEXT
```



```

490 COLOUR1
500 PRINT"GCOL";gcol1%;",",col1%
510 COLOUR2
520 PRINT"GCOL";gcol2%;",",col2%
530 COLOUR3
540 PRINT"PRESS CTRL-@
550 REPEATUNTILINKEY(0)=0
560 ENDPROC

```

## softspot – tennis/walltrap

These are games of bat and ball for two players. Both use the bat in the same way as Wallball (see February issue), so if you have mastered the technique of moving the bat in that game you should find these of interest. Tennis is fairly straightforward. Start the ball by pressing the fire button. The score you are aiming for is 21, and you get a point each time your opponent allows the ball past him to hit the rear wall.

Walltrap is enormous fun for two people who have both mastered the technique for moving the bat at an angle and placing the ball where they want to. Beginners are likely to find it frustrating. You are aiming for 50, and get a point each time the ball bounces off the end wall or the red wall on your opponents side.

```

L.
10 REM   TENNIS/WALLTRAP
20 REM   by Peter Voke
30 :
40 REM   December'82
50 :
60 REM   Version 1.1
70 :
80 REM Takes up 3.10k memory
90 :
100 REM   Requires 32k
110 :
120 REM   Written on OS 0.1
130 :
140 REM   (c) LASERBUG 1983
150 :
160   :::::
170 :
200 MODE 1
210 PROCGAME
220 MODE 2
230 PROCSCREEN
240 PROCSTART
250
260 REPEAT
270   PROCPLAY
280   UNTIL S%>R% OR T%>R%
290
300 PROCMORE
310
320 END
330
340
350 :::::
360
370 DEFPROCBAT
380
390 GCOL3,2
400 IX=KZ: JZ=LZ
410 KZ=528-ADVAL(1)DIV2048*16
420 LZ=FNJOY(2)
430 MOVE IX,JZ: DRAW IX,JZ+96
440 MOVE KZ,LZ: DRAW KZ,LZ+96
450
460 DZ=FZ: EZ=GZ
470 FZ=1232-ADVAL(3)DIV2048*16
480 GZ=FNJOY(4)
490 MOVE DZ,EZ: DRAW DZ,EZ+96
500 MOVE FZ,GZ: DRAW FZ,GZ+96
510
520 ENDPROC
530
540
550 :::::
560
570 DEFPROCEND
580
590 SOUND1,-13,200-C%*12,2
600 IF XZ<600 TZ=TZ+1 ELSE SZ=SZ+1
610 PROCSCOR
620
630 ENDPROC
640
650
660 :::::
670
680 DEFPROCZAP
690
700 IF XZ<600 VZ=VZ+(FNJOY(2)-JZ)DIV10
710 IF XZ>600 VZ=VZ+(FNJOY(4)-EZ)DIV10
720 IF ABS(VZ)>Max% VZ=Max%*SGN(VZ)
730 SOUND0,-9,4,1
740
750 ENDPROC
760
770
780 :::::
790
800 DEFPROCSCOR
810
820 VDU4
830 COLOUR 5
840 PRINT TAB(0,2) SZ
850 COLOUR 2
860 PRINT TAB(14,2) TZ
870 VDU5
880
890 ENDPROC
900
910
920 :::::
930
940 DEFFNJOY(QZ)
950
960 HZ=ADVAL(QZ)DIV64
970 IF HZ<96 HZ=96
980 IF HZ>803 HZ=803
990
1000 =HZ
1010

```

```

1020
1030 ::::::::::::::::::::::::::::::::::::
1040
1050 DEFPROCPLAY
1060
1070 PROCBAT
1080
1090 FOR Q%=1 TO 3
1100   Q%=X%: P%=Y%
1110   X%=X%+U%: Y%=Y%+V%
1120   Z%=Y%+V%: IF Z%>911 OR Z%<112 V%=-V%: SOUND2,
1,50,5
1130   AZ=Y%-L%: Z%=Y%-(L%+J%)DIV2
1140   IF (Z%>115 OR Z%<0) AND (AZ%>115 OR AZ%<0) THEN
1160
1150   IF X%>I%-64 AND X%+U%<K%-32 U%=ABS(U%): X%=K%
-32: SOUND0,-9,4,1: PROCZAP
1160   AZ=Y%-G%: Z%=Y%-(E%+G%)DIV2
1170   IF (Z%>115 OR Z%<0) AND (AZ%>115 OR AZ%<0) THEN
1190
1180   IF X%<D% AND X%+U%>F%-32 U%=-1*ABS(U%): X%=F%
-64: SOUND0,-9,4,1: PROCZAP
1190   M%=X%+48+48*SGN(U%)
1200   C%=POINT(M%,Y%-10)
1210   IF C%<2 U%=-U%: PROCEND
1220   GCOL3,3: MOVE0%,P%: VDU226: MOVE X%,Y%: VDU22
6
1230   NEXT
1240
1250 ENDPROC
1260
1270
1280 ::::::::::::::::::::::::::::::::::::
1290
1300 DEFPROCSCREEN
1310
1320 VDU23;8202;0;0;0;
1330 VDU23,226,2,5,5,5,2,0,0,0
1340 ENVELOPE1,129,-2,8,0,10,40,0,127,0,-20,-30,100,
100
1350 @%=&404
1360
1370 GCOL0,4
1380 MOVE 0,900: MOVE 1279,900
1390 PLOT 85,0,95: PLOT 85,1279,95
1400
1410 GCOL0,7
1420 MOVE 639,900: PLOT 21,639,95
1430 MOVE0,908: DRAW1279,908
1440 DRAW1279,84: DRAW0,84: DRAW0,908.
1450
1460 IF Game%=0 THEN 1520
1470 GCOL0,1: MOVE600,900: MOVE679,900
1480 PLOT85,600,640: PLOT85,679,640
1490 MOVE600,95: MOVE679,95
1500 PLOT85,600,360: PLOT85,679,360
1510
1520 COLOUR 3: IF Game% PRINT TAB(6,2) "WALLTRAP" EL
SE PRINT TAB(7,2) "TENNIS"
1530 PRINT TAB(0,30) " (c) LASERBUG 1983"

```

```

1540 SZ=0: TZ=0: PROCSCOR
1550
1560 ON ERROR GOTO 300
1570
1580 ENDPROC
1590
1600
1610 ::::::::::::::::::::::::::::::::::::
1620
1630 DEFPROCSTART
1640
1650 K%=-100: LZ=0: F%=-100: G%=0
1660 VDU5: PROCBAT
1670 REPEAT: PROCBAT: UNTIL ADVAL(0) AND 3
1680 X%=0: Y%=512
1690 GCOL3,3: MOVE X%,Y%: VDU226
1700 U%=32: V%=SGN(RND)*5
1710 Max%=20
1720
1730 ENDPROC
1740
1750
1760 ::::::::::::::::::::::::::::::::::::
1770
1780 DEFPROCGAME
1790
1800 ON ERROR RUN
1810 VDU23;8202;0;0;0;
1820 VDU19;2;0;19,7,0;0;
1830 PRINT TAB(10,9) "Choose your game: """" TAB(13)
"A. Tennis"" TAB(13) "B. Walltrap"
1840 PRINT"" TAB(13) "Press A or B"
1850 IF INKEY-66 Game%=0: R%=20: ENDPROC
1860 IF NOT INKEY-101 THEN 1850
1870 Game%=-1: R%=49
1880
1890 ENDPROC
1900
1910
1920 ::::::::::::::::::::::::::::::::::::
1930
1940 DEFPROCMORE
1950
1960 VDU4: COLOUR 3: PRINT TAB(0,30) " Another game?
Y/N ";
1970 IF INKEY-69 RUN
1980 IF NOT INKEY-86 THEN 1970
1990 CLS: *FX15,0
2000 CLEAR: MODE7
2010
2020 END

```

## pie charts

You might have an idea for a program but not be exactly sure how to approach the topic – or your maths might be a bit shakey!

Below is the basic outline for a program to draw a pie chart. The actual program does work and work correctly. It does need a good deal of expanding though before it is ready to be used. For starters the presentation is terrible and only 4 items can be entered. Try and work on the program and send us your efforts.

```

>UIL.
 10REM PIE CHARTS
 20REM by Paul Barbour
 30:
 40REM Version 0.1
 50REM (Designed to be expanded)
 60:
 70REM (c) LASERBUG 1983
 80:
 90  :::::
100:
110MODE7:VDU23;8202;0;0;0;
120PROCdata_in
130PROCprocess
140MODE2:VDU23;8202;0;0;0;
150PROCdraw
160PROClabel
170REPEATUNTIL0
180:
190  :::::
200:
210DEFPROCdata_in
220PRINT""HOW MANY ITEMS (Max.4)"
230INPUTitem%
240IFitem%<10Ritem%>4THEN220
250DIMdata%(item%)
260FORinfo%=1TOitem%
270PRINT""Enter value of item ";info%";. ";
280INPUTdata%(info%)
290data%(info%)=ABS(data%(info%))
300NEXT
310ENDPROC
320:
330DEFPROCprocess
340sum%=data%(1)+data%(2)+data%(3)+data%(4)
350DIMangle%(item%)
360FORprocess%=1TOitem%
370angle%(process%)=(data%(process%)/sum%)*360
380NEXT
390circle%=0
400ENDPROC
410:
420DEFPROCdraw
430PROCcircle(1,angle%(1))
440PROCcircle(2,angle%(2))
450PROCcircle(3,angle%(3))
460PROCcircle(4,angle%(4))
470ENDPROC
480:
490DEFPROCcircle(col%,ang%)
500VDU29,640;512;
510GCOLOR,col%
520FORcircle%=circle%TO(circle%+ang%)
530MOVE0,0
540DRAW SINRAD(circle%)*400,COSRAD(circle%)*400
550PLOT85,SINRAD(circle%+1)*400,COSRAD(circle%+1)*400
0
560NEXT
570ENDPROC
580:

```

```

590DEFPROClabel
600PRINTTAB(0,0);
610FORlabel%=1TO4
620COLOURlabel%
630PRINT"ITEM ";label%
640NEXT
650ENDPROC

```

Paul Barbour

## cursor fun

The BBC Micro's cursor is very neat but at times you might wish that it was not flashing or bigger or flashing faster. It is possible to alter the cursor in these ways and the way you do it is by using VDU23. Rather than list out all the possibilities which would take up rather a lot of space below is a short program to alter the cursor for you. It is very easy to follow but as it is designed to show you a principle it has no error trapping. You might find it very useful to have several different cursors for different parts of the program i.e. in a data handling program one cursor when you are entering data, another one when you edit the data, etc. Anyway, the below might give you some good ideas . . .

```

L.
 10 REM CURSOR FUN I
 20 REM by Paul Barbour
 30 :
 40 REM Version 1.0
 50 :
 60 REM (c) LASERBUG 1983
 70 :
 80  :::::
 90 :
100 CLS
110 VDU23,255,-1,-1,-1,-1,-1,-1,-1,-1
120 INPUT"Which mode ",mode%
130 MODEmode%
140 FORcursor%=0TO255
150  CLS
160  VDU23;8202;0;0;0;
170  PRINT"VDU23,0,10,";cursor%";,0;0;0;"
180  PRINTTAB(0,4);"Press RETURN"
190  IFmode%=7THENPRINTTAB(0,2);CHR$151;CHR$255;:EL
SEPRINTTAB(0,2);CHR$255;
200  VDU23,0,10,cursor%,0;0;0;
210  REPEAT=GET
220  UNTIL=13
230  NEXT

```

## tip time – \*FX in programs

If in a program you have wanted to use several different values in a \*FX statement then you will know that a procedure like:

```

>L.
10000DEFPROCfx(A%,B%,C%)
10010*FXA%,B%,C%
10020ENDPROC
>

```

is illegal. The way to do \*FX in programs where you want the values to change is to load the appropriate registers with the values you want and to use the OS subroutine OSBYTE. If this sounds too confusing we have already made up a suitable PROCedure for you:

Recently we have been receiving a large amount of mail with regards to Dr. Susans article *Conversions for RGB Inputs* (LASERBUG Issue 5, October '82 pages 28/9) which described how to convert a Ferguson Portable (TX10 chassis) to take RGB Inputs. In this country it is the TX9 chassis that is mainly used in these models. Surprisingly it is not Dr. Susans who wrote the article who have been receiving the correspondence, nor ever LASERBUG but David Prideaux who successfully carried out the alteration and wrote in to thank Dr. Susans (LASERBUG Issue 8, January '83, pages 4/5). To try to stop some of the strain on David Prideaux below is a follow up from David Prideaux on the modification he performed. We will still pass on your letters to David of course but hopefully the article below might help you a bit more.

**Ferguson 3755 portable**

The circuit I used is identical to that suggested by Dr. Susans in LASERBUG except for some resistor value changes (see diagram). The components mount on a small 0.1" matrix board which I have fixed to the inside of the cabinet on the left hand side (from the back) with double-sided adhesive pads. It would be possible with a little ingenuity to mount it much closer to the I.F. module on the left hand side, next to the U.H.F. tuner module) and thus keep the connecting leads shorter, minimising interference problems on T.V.

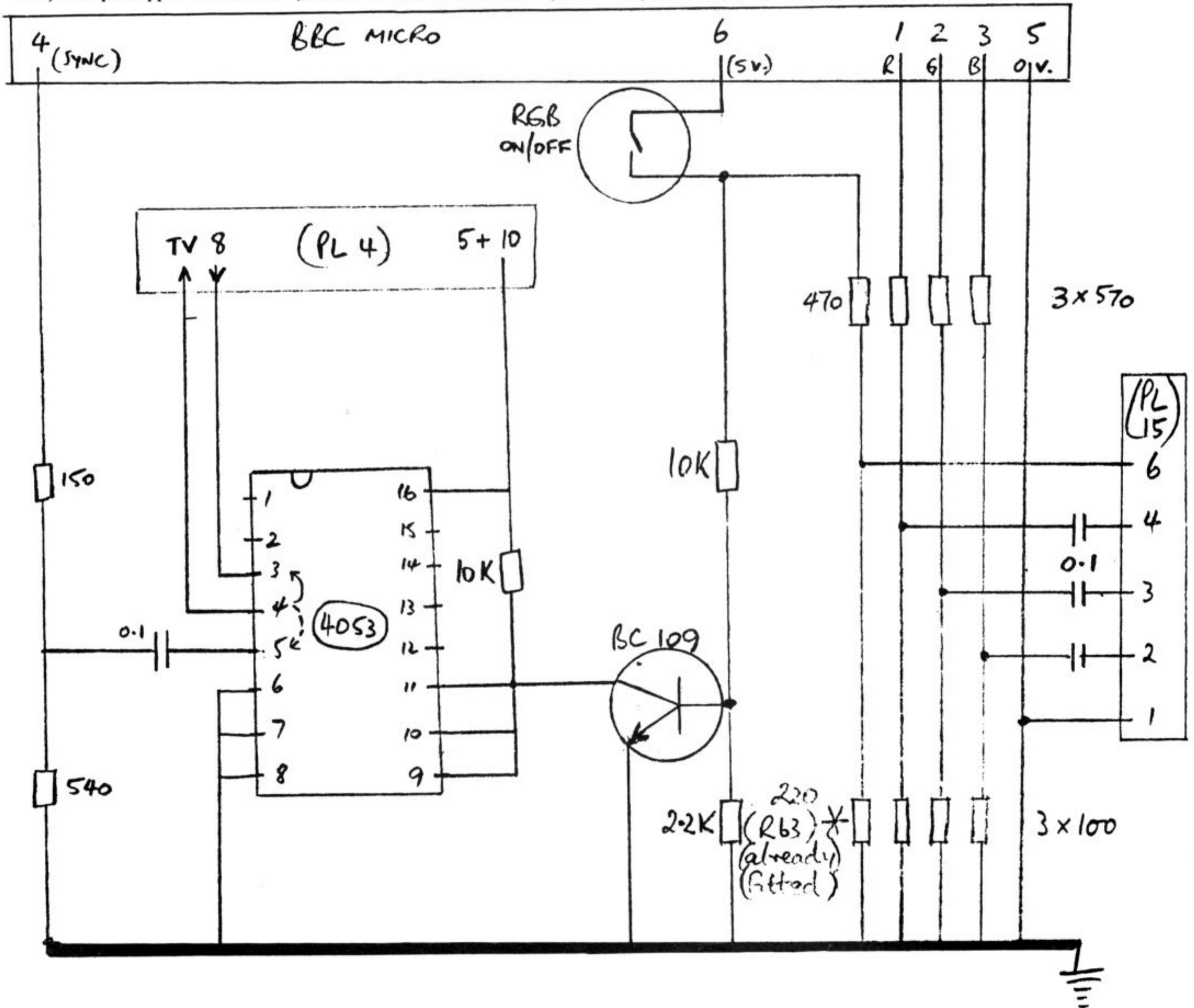
The convenient test points Dr. Susans refers to on the TX 10 do not exist on the TX 9, but I found that the circuit connections required can be picked up on the edge connector (PL 4) of the I.F. module, which unplugs from the main P.C.B. after unhooking the spring retaining clip. This has the advantage that the connections can be made out in the open. The only awkward one is pin 8 which has to be unsoldered and then bent out of its hole in the module P.C.B. so that a lead can be soldered to each side of the break, i.e. one to the pin, one to the hole. The other solder connections can be made to convenient points on the module P.C.B. tracks. A length of multicore cable connects my board to PL 15, the Teletext plug on the set (directly under the tube). The pins appear to be at 0.2" pitch so alternate holes in a 0.1" pitch

connector socket (e.g. RS 467-633 + 467-596) do the job. Obviously, permanent solder connections could be made instead if you prefer.

I have inserted a small toggle switch to switch between T.V. and RGB. As I was trying not to cut holes in the T.V. I brought this and the lead to the computer, together with the second mains lead, out through the aperture left by removing the left-hand carrying grip, and I attached the switch with more magical double-sided pads. If you have small children around you will obviously have to think of another way, as you will not want to leave such a large tempting hole in the receiver case.

Finally the most awkward part of the job. It is of course essential to earth the chassis, which entails also supplying the set through a mains isolating transformer. Because of the relatively large current (about 3 amp.) drawn by the degaussing coil on switch-on, a very large transformer would be necessary. However to keep the size to a sensible level, the degaussing coil can be supplied from a separate mains supply. This entails the only soldering work on the main P.C.B., to remove the Zener diode Z 2, located in the far left-hand corner of the board. I then mounted this on a separate small piece of matrix board and brought its connections to two P.C.B. pins (e.g. RS 433-860) positioned so that PL 1, the connector to the degauss coil, simply plugged onto the pins instead of its original socket on the main P.C.B. The whole board was then well wrapped with insulating tape before sticking to the side of the cabinet. The mains lead for this supply is 3-core, so the earth conductor can be used as the chassis earth and the original T.V. mains supply lead (2-core) can be used for its new isolated supply.

My isolating transformer is in a metal case so I have mounted two different sockets on it (2 amp and 5 amp round-pin) so that they cannot be inadvertently interchanged. One is the isolated output for the T.V., the other un-isolated for the degauss supply. There are two limitations on this arrangement – one is that on/off control MUST be by means of the main switch to the transformer. If you use the T.V. on/off switch the degauss coil is still 'live'. Also the 'degauss' plug must never be unplugged, or the earth connection will be lost. You can obviously modify this arrangement to avoid any chance of danger if you feel this is necessary.



The article below was sent in totally anonymous. The writer thought that if he supplied his name and address his own chance of winning one of the prizes were at risk. The writer has himself solved the Castle of Riddles and expects to have done well in the competition. Just in case you don't know about the Castle of Riddles I'll explain. Castle of Riddles is an adventure program from Acornsoft. The prizes for solving said adventure program amount to over £3000. For the rest I'll hand you over to King Of The Ring (?!?!?) . . .

At the time of writing, Castle of Riddles is just six days old, but by the time you read this, almost a month of this existing treasure hunt will have gone, and there may be only a week or two left to the closing date. If you have not solved Castle already, you have probably given up any idea of getting first, second or third prizes but perhaps would still like that badge saying "I solved . . ."

If you haven't bought Castle of Riddles, don't be put off by the somewhat sordid commercial aspect. It is an excellent adventure game, tough, challenging, hilarious in parts, and finally exhilarating when you solve it. A few bugs (mentioned below) detract it from an otherwise excellent package.

If you are stuck, here are a few hints to help you get that badge (exclusive to LASERBUG readers, of course!). First, for the absolute beginners: the lamp is easily found north of the castle, and there is something you need in the forest, but only one thing. If you get stuck in a maze of white rooms, try turning the lamp off for a second. Once you reach the courtyard in the centre of the castle, this is a good place to type SAVE. You can then try exploring north, east, south and up the stairs without losing too much ground. Your one squirt from the Fountain of Youth should really be saved as you will need it later.

For the more advanced, the following may help. If you get totally lost in a maze of black tunnels, try bringing a black rod, which you can find elsewhere in the adventure and which spouts white smoke when you wave it, into this maze, and wave it around. This will allow you to "map" the maze, and even give a hint of the spell word that spirits you out of the maze exit when you finally find it. (What **order** should the colours be in?)

If you can't get passed the small happy bear, just persist. If you don't know how to get the rod out of the west sector, try SOS when you see dots and dashes. To retrieve the guarded necklace, find something to scare away the bear, then go back. There is a bug associated with the bear, but not a serious one. Another bug is more useful. In the initial position, type RUB LAMP, or RUB BOX and see what happens. That should help anyone stuck in front of a large mirror. Incidentally, if you keep getting killed by a nasty bloodsucking vampire in front of that mirror, try **fooling** it with something like blood.

The giant's shooting gallery you will have to find on your own way through. All I will say is that there is only one safe move at any one time, and when you get in to the brooch the second time you have to stop there for two moves. Ultimately, you must leave a couple of things here in order to recover the brooch and to get inside a box, rather than merely waiting, or picking up the brooch.

If you find you need a hanglider on two occasions but only have one available, you're getting very close to winning. Now think: what else could you use to break a fall from a great height? If you are running out of battery, the Fountain can be used to regenerate the lamp as well as put it out. When you finally win, you will find yourself in possession of a rather tatty cushion. I think the game writer intended you to revivify this in the Fountain too, but in fact it makes no difference to the score at all: this seems to be a third bug in the program, and the most serious.

The fourth bug is in the boat floating on the flood. The computer seems to react somewhat irrationally here, but if you have a bucket with you to keep the boat afloat you should get through. Oh, if you can't get that bucket out of the mud, just try persistence again ("What I tell you three times is true".)

That is just about all the hints you need. The one spell word given you in the game opens the safe: the other spell word is needed in the exit of the black maze, and is in my opinion the toughest conundrum set by this excellent adventure. If the smoke from the rod doesn't give you the hint, think of a micro that definitely isn't as good as the BEEB! Somebody at Acornsoft really has a sense of humour.

So, the best of luck. I hope these hints have not annoyed any purist adventurers – you didn't **have** to read the article, though. Finally I must thank Andy and Pandy, without whom I should not be, (very hopefully).

King Of The Ring

continued from page 19

>L.

9990REM \*FX AZ, XZ, YZ

10000DEFPROCfx (operation, value1, value2)

10010AZ=operation:REM Set accumulator

10020XZ=value1:REM Set X register

10030YZ=value2:REM Set Y register

10040CALL&FFF4

10050ENDPROC

Paul Barbour

## EPSON'S LATEST PRINTERS

SAVE £70-£80 (+VAT) when you order your superb new FX80 or MX100 printer from Datatech, the Epson specialists.

We also have large stocks of Epson sundries including cartridge ribbons, dust covers, fanfold paper, continuous labels, etc.

For FREE BROCHURE and special DISCOUNT OFFER write now to:

**DATATECH LTD (LB)**  
3 BRAMHALL CLOSE, TIMPERLEY,  
ALTRINCHAM, CHESHIRE, WA15 7EB

## classified

Got something to sell or buy? If so then LASERBUG classified Ads are for you. Rates are 10p/Word (minimum 15 words) or £10 per column centimetre (minimum 2 centimetres). Send your cheques/PO's made payable to LASERBUG to Classified Ads, LASERBUG, 10 Dawley Ride, Colnbrook, Slough, Berks., SL3 0QH. We cannot guarantee your ads will go in any particular issue but will do our best to ensure a prompt service.

### BBC SOFTWARE

## FOOTBALL POOLS PREDICTOR

This is a very powerful, but easy to use, mathematical and statistical forecasting program. It uses SIX different methods of prediction and even allows you to combine your own judgement with its forecasts. No separate 'database' is required and you don't need to keep any special records. It runs on BBC Model A or B and is supplied on cassette with full instructions.

£4.99

**MAYDAY SOFTWARE**  
181 PORTLAND CRESCENT  
STANMORE, MIDDLESEX HA7 1LR

BBC MODEL A/B DISASSEMBLER as reviewed in this issue. On tape with manual for £5 inc. From David T. Knell, 13 Northumberland Road, Leamington Spa, Warwickshire CV32 6HE. Tel. (0926) 24705 between 6 and 7 p.m.

The Computer Programme II – Making The Most Of The BBC Micro  
Presented by Ian McNaught David  
With Patrick Eagar and Malcolm Rivers  
Episode II – Getting Down To BASIC

The program opens with Patrick Eagar, a world famous cricket photographer, talking about his first experiences with computers. He bought a ZX81 which showed him how useful a computer could be to his job. He obtained an Apple II and wrote his own business software to suit his special needs.

Then we move back to Mac in the studio. He starts up his computer and loads in a program called sticks which he says draws a picture looking like “a trajectory of a chopstick thrown up in a Chinese Restaurant” which leads very smoothly into the next scene.

This is a Chinese Restaurant which is used as an example of what you do when programming – you speak to the waiter in English and he writes down your order in Chinese i.e. converts what you say into a language that he understands. This is given to the cook and he, understanding the chinese cooks your meal and in due course you receive what you ordered, all without having to understand what was really going on inside the workings of the restaurant. In the same way on your micro you program in BASIC which is converted into a machine code (the chinese writing) which is acted upon by the CPU 8 the chef) and you get out what you requested the computer to do (your meal).

Using a teacher and his children as an example then Mac shows how to write a program. The teacher thinks of two numbers, multiplies them together and remembers the answer. He then asks the child the question and answers right or wrong. Back in the studio Mac gives you a flowchart of what the teacher was doing and converts this into a program. Trays with pieces of cardboard represent variables with their contents and a dice is used for a random number. The viewer is shown the finished program working.

The situation is then expanded by giving the child three chances to answer the question and the program is likewise altered.

The final scenes show the BBC Micro used in schools. A class project on Treasure was tied in with the Kingdom program. Several other educational programs were shown and all these, we were told, will be available on the telesoftware service. Predictably the children were very un-inhibited about using the computer whereas the teachers were very apprehensive.

This episode was important in so far as it actually introduced programming showing a real life example and how easy it is to transfer this into a computer program. A good start in the programming field and should show people that computer programming isn't quite as hard as it may seem.

Paul Barbour

## micronet 800

With the extensive publicity campaign they have launched and the many articles in the computer press you couldn't have failed to hear about Micronet 800 which is hoped to be the making of Prestel.

To use Micronet you have to buy a special modem directly from Micronet for £49.95 + VAT along with the software which is included in the package. One of the first computers that a Micronet adaptor is available for is the BBC Micro. After negotiations between ourselves and Micronet we have decided to become part of that system.

At the moment LASERBUG is involved in Micronet 800 in three ways, firstly in the area of telesoftware. A good number of the programs from the back issues of LASERBUG (24 in all at present) will be reproduced on Micronet in the form of telesoftware. This will mean that if you are a subscriber to Micronet, instead of having to type in a program from a back issue you will be able to load it directly. All programs from the magazines will be what is known as Public Domain Software which means that you will be able to access them free of charge. Only programs from old magazines will be appearing on Micronet so that subscribers still get first look at all programs.

Secondly on Micronet there will be about 12 pages of promotional information giving a description of LASERBUG and trying to persuade people to join.

Thirdly we will be having our own newsletter on Micronet. This will be at least 50 pages long.

At present though most of this still has to be sorted out completely. At the time of writing Micronet still wasn't even launched – it is (or was for you) due to be launched on the 21st February. At this time only the programs will be there. We will soon be sending Micronet copy for the promotional pages but what will appear there we still have no idea.

Your comments and views on Micronet or the LASERBUG pages on Micronet would be appreciated. Also any suggestions as to what will go into the LASERBUG magazine on Micronet.

As we don't know much about the exact details yet I can't give you any page numbers to look at, but I'm sure you will find us if you look in one of the

indexes.

You can also contact us through the Prestel electronic mail box or something like that but as I'm just as much in the dark about Prestel/Micronet as most of you don't ask me to explain. Once things have got going we will do an article on Micronet and the service it provides.

Paul Barbour

## competition

This months competition is to crack the code below. The code represents a sentence about your favourite user group and needs a little bit of thinking about. A look at page 199 in the user guide might help you **EXCEPT** that 199 is also a code, all be it a much simpler one. So look at the numbers below and send the sentence it represents and how you decoded it, preferably including the program you used to display the sentence. If you are really stuck decoding the number 199 and looking at the suitable page will give you your answer. Final entry date is the 15th April. Oh yes, the code. Ready? Wellm try and decode this then:

7665836982668571327079823284  
7269326666673277736782794613

## computer conversions part IV

**TAB:** Standard BASIC

**TAN:** Gives the tangent of its radian argument (this is included in many micro's BASIC whereas SIN and COS are not because it is possible to work out sines and cosines with just tangents).

**THEN:** Part of the IF condition. Standard BASIC.

**TIME:** Pseudo variable unique to the BBC Micro holding the value of the elapsed time clock. All micro's in general have an accessible clock – some use another pseudo-variable such as TIS or just TI. On others it must be accessed by using PEEKs.

### BBC SOFTWARE

**EDUCATIONAL – 1** **A or B** **£8.05**  
Hours of fun and learning for children aged 5 to 9 years. Animated graphics will encourage children to enjoy maths, spelling and telling the time. The tape includes **Math 1, Math 2, Cubecount, Shapes, Memory** (Model B only), **Spell and Clock**.

**EDUCATIONAL – 2** **A or B** **£8.05**  
Although similar to Educational – 1 this tape is more advanced and aimed at 7 to 12 year olds. The tape includes **Math 1, Math 2, Area, Memory** (Model B Only), **Cubecount and Spell**.

**GAMES OF LOGIC & CUNNING** **A or B** **£9.20**  
For children and adults alike. The tape includes **Auction, Flip, Reverse, Telepathy and Rotate**.

**SUPERLIFE** **B** **£9.20**  
Fast (machine code) version of a popular 'Game of Life' in a large universe. Can you produce 'Blinkers', 'Spinners', 'Gliders' and 'Spaceships' or have you only wondered what they look like? All this in Superlife and more as this tape includes 'Competitive Life' with the Reds and the Blues competing for space; perhaps you can alter their evolution.

**KATAKOMBS** **B** **£9.20**  
Are you cunning enough to discover and seize the treasure in the Katakombs AND return alive? What and where are your enemies? Can you outwit them? Yes? Then your adventure will take you through unending forests, besides tumbling streams, over lonely plains to desolate ruins and finally underground to the tortuous Katakombs. Be prepared for anything!

**UTILITIES** **A or B** **£8.05**  
Behind the mundane title lies an assortment of useful procedures and functions which can save you hours/days of programming effort:- date conversion, input and validation routines, graphic routines (cube, rectangle, etc.), sorts, search and many more.

★★★ **SPECIAL OFFER** ★★★ *Any three cassettes for £20.70*

Add 50p p/p per order.  
Please state your Model & Ref. LA 1.  
Cheques/P.O. payable to Golem Ltd.,  
77 Qualitas, Bracknell, Berks. RG12 4QC.  
Tel: (0344) 50720

**TO:** Part of the FOR loop. Standard BASIC.  
**TOP:** Pseudo-variable holding the first memory location after the current program. Unique to the BBC Micro – must be found out on most other micros by using PEEKs.  
**TRACE:** This command makes the computer prints out the line numbers as it executes them. TRACEON turns on the facility and TRACEOFF the opposite. On other micros TRACEON could be replaced by just TRACE or TRON (obviously before the film!). It could be switched off by TROFF or NOTRACE depending on the machine.  
**TRUE:** A fixed variable holding the value of -1. Unique to the BBC Micro.  
**UNTIL:** Part of the REPEATUNTIL loop. Unique in BASIC to the BBC Micro.  
**USR:** Fairly standard BASIC which calls an assembler subroutine which returns a single value. The A, X and Y registers must be set before it is called.  
**VAL:** Standard BASIC.  
**VDU:** Unique to BBC Micro. Provides another way of accessing the VDU drivers apart from PRINT – VDU is more versatile than print for certain uses.  
**VPOS:** Unusual BASIC. A pseudo variable returning vertical position of cursor (VTAB on certain machines).  
**WAIT:** A command used to suspend the machines operation for a specified length of time. Implemented on quite a few machines (not BBC).  
**WHILE/WEND:** Alternative version of REPEATUNTIL loop as the syntax is reversed i.e. X=1:REPEATX=X+1:UNTILX=10 becomes X=1:WHILEX<10:X=X+1:WEND.  
**WIDTH:** Limits the print field either on screen or printer to a specified number of characters (we use WIDTH34 at LASERBUG for program listings).  
**WRITE:** Writes data to file. On BBC Micro use PRINT#.  
 (Sorry that we don't go up to Z but nobody has invented a BASIC with a command going that far – unless you know different.)

Paul Barbour

©LASERBUG 1983. Reasonable care is taken to avoid errors in this magazine. However, no liability is accepted for any mistakes which occur. The opinions expressed in this magazine are those of the authors of each individual article and do not necessarily represent those of the Editor. No material may be reproduced in any way whatsoever (for any reason) without the written consent of the Editor.

LASERBUG is edited by Paul Barbour.  
 The contributors this month were Paul Barbour, Nick Goodwin, Peter Voke and Patrick Dowling.  
 The final copy date for each issue is the beginning of the month preceding that of the cover date.  
 Articles and programs are always welcome. Please make sure that your work is original and has not been copied from elsewhere nor submitted to any other organisation. Payment for articles is at the rate of £5 or £10 per contribution depending on content, etc. Other rates can be negotiated for work of high standard. All contributions should be typed or computer printed with double spacings and at least a 1" margin. Hand written material may be subject to delay and error. All programs longer than 10 lines should be submitted on cassette both at 1200 and 300 baud. If a listing is supplied the computer should be set to WIDTH34 and LIST07 for 80-column listings. The first lines of the program should be REMed in the normal LASERBUG standard.  
 All articles published in LASERBUG become the property of LASERBUG, as does the copyright.  
 LASERBUG is typeset and printed by Prestige Press (UK) Ltd., East Street, Chesham, Bucks.  
 All correspondence should be sent to LASERBUG, 10 Dawley Ride, Colnbrook, Slough, Berks., SL3 0QH.  
 If in difficulty phone 02812-3064 during office hours only.  
 Membership to LASERBUG:  
 UK 12 month membership ..... £12.00  
 UK 6 month membership ..... £6.00  
 (Overseas memberships 12 months only)  
 Overseas Surface Mail Membership ..... £14.00  
 Postal Band A Air Mail Membership ..... £17.00  
 Postal Band B Air Mail Membership ..... £19.00  
 Postal Band C Air Mail Membership ..... £20.00  
 Forces Membership ..... £12.00  
 Forces Outside Europe Air Mail Membership ..... £14.00  
 Advertising rates available on request.

continued from page 13

```
2330 ENDPROC
2340
2350 DATA 1,1,0,0,0,6,3,3,30,-4,-1,-1,120,80,1,1,89,
7
2360
2370 IF ERR=17 GOTO 290
2380 IF ERR>200 CLOSE#X:PRINTTAB(0,30)"Tape error, pr
ess any key.":G=GET:GOTO290
2390 VDU7,31,0,29:REPORT:PRINT" @ line no. ";ERL
```

Patrick Dowling

LASERBUG  
 10 Dawley Ride,  
 Colnbrook,  
 Slough, Berks,  
 SL3 0QH.



SUBSCRIBE NOW AND GET THE MOST OUT OF YOUR BBC MICRO !!!

Please enrol me as a member of LASERBUG. I enclose a cheque/postal order for £ .....  
 for a ..... subscription (for details of the  
 different memberships please see Credits on the inside back cover)

NAME .....

ADDRESS .....

..... DATE .....

MEMBERSHIPS CANNOT BE BACKDATED – PLEASE ORDER BACKCOPIES SEPARATELY FROM THE SPECIAL OFFERS SECTION

I would/would not like my name and address printed in the Contacts page of LASERBUG

Please make all cheques/postal orders payable to LASERBUG

NOTE: When your membership application is received you will be sent the current copy of the magazine along with membership card. A reminder to rejoin will be sent with the penultimate magazine in your subscription.

# LASERBUG SPECIAL OFFERS

## MEMBERS ONLY DISCOUNTS:

If you are a member of LASERBUG you will have a membership card that not only proves that you belong to LASERBUG but also entitles you to special discounts that certain companies have offered. At the present time Dracal Ltd. will give you a 5% discount and Futura Software a 25% discount on all their goods. Dracal Ltd. produce many different things for the BBC Micro including design sheets monitor stands. Full details from Dracal (North West) Ltd., P.O. Box 130, Warrington, WA1 4QB. Phone Padgate 815419. Futura Software produce a large number of games for the BBC Micro to which you can get a 25% discount on! More details from Futura Software, 63 Lady Lane, Chelmsford, Essex, CM2 0TQ. Details on both these companies products were given in issues 6 and 9 respectively. Please quote your membership number both when corresponding and ordering goods from these companies. We are not acting as agents for the two companies concerned and cannot be held responsible for their actions or services in any way whatsoever.

**STOP PRESS:** Another company has just contacted us offering you a 10% discount. This time it is Silent Computers of Tottenham. They produce a nice little console for the BBC Micro made of aluminium which you can put your BBC Micro, rest of monitor on top and still have room for either dual drives or a single drive and a 2nd processor/teletext. They are soon to produce an extra bolt on module to allow another peripheral in the same structure. The price is £39.99 plus £4.00 P & P but if you quote your membership number you can get the console for £36.00 plus £4.00 P & P. More details from Silent Computers, 27 Wycombe Road, Tottenham, London, N17 9XN.

## OTHER LASERBUG SERVICES AND GOODS:

### LASERBUG DUST COVERS FOR YOUR BBC MICRO ONLY £3.25

LASERBUG Dust Covers are manufactured for us by a well known company and are made to extremely high quality. Full details on the covers were given in issue 5 but briefly they are made of polyester/cotton in a fawn colour and fit snugly over your BBC Micro. Each Dust Cover is £3.25.

### BACK COPY SERVICE

Missed out on a copy of LASERBUG??? If so then use our back copy service and catch up on those issues you missed. We have stocks of all magazines except issue 1. Brief details of each issue available are:

- Issue 2 (16 pages) – Sound/Envelope, \*FX Part I, Teletext Part II, User Definable Keys, Pontoon
- Issue 3 (24 pages) – Programmers Corner, \*FX Part II, Epson Screen Dump, Softreview, Telesoftware
- Issue 4 (20 pages) – Hardspot, What Printer?, Bookreview, Oddspot, Pixel Power, Moving Things
- Issue 5 (32 pages) – Wordprocessor, Machine Code, Disassembler, Seikosha Screen Dump, RGB Conversions
- Issue 6 (24 pages) – Software Protection Part I, Puzzle Program, How To Use Joysticks, Queryspot
- Issue 7 (24 pages) – Software Protection Part II, Alphabet Tester, Questionnaire Results, Club Reports
- Issue 8 (24 pages) – Memory Analyser, Assembler Programming On The BBC Micro Part I, Diskspot
- Issue 9 (24 pages) – Wallball, ADC Corner, Make The Most Out Of Sound, Grand Prix, Nine Dice

Each back copy costs £1.25.

### PHOTOCOPY SERVICE

Desperately want an issue 1??? Well, as we said above we have no more of this issue in print but we can offer you a photocopy of it for £1.60. Also if you want a copy of a particular article but do not want to pay £1.25 for the whole magazine we will photocopy any article in LASERBUG for you for 50p per article. You should remember that LASERBUG is copyrighted and as such it is illegal to photocopy any part of it without written permission.

### CHEAP BOOKS FROM INTERFACE

Interface have kindly allowed us to offer you a discount on both books that they publish for the BBC Micro. The first is Making The Most Of Your BBC Micro by Tim Hartnell and the other The BBC Micro Revealed by Jeremy Ruston. The books cost £5.81 and £7.16 respectively. Both have been reviewed in LASERBUG.

### LASERBUG CASSETTE LEADS ONLY £3.00 EACH

If you want a cassette lead for your BBC Micro you could be paying between £4.00 and £8.00 – why bother when LASERBUG can supply you with what you need for £3.00. We can offer you 2 leads, lead A which is a 7-pin DIN to 1 DIN/1 2½mm Remote Jack or lead B which is a 7-pin DIN to 2 x 3½mm jacks and 1 x 2½mm remote.

- Please send me . . . . . LASERBUG Dust Cover(s) @ £3.25 each  
 . . . . . Back Copy(s) of LASERBUG Issue(s) . . . . . @ £1.25 each  
 . . . . . Photocopy(s) of LASERBUG Issue 1 @ £1.60 each  
 . . . . . Copy(s) of Making The Most Of Your BBC Micro by Tim Hartnell @ £5.81 each  
 . . . . . Copy(s) of The BBC Micro Revealed by Jeremy Ruston @ £7.16 each  
 . . . . . LASERBUG Cassette Lead(s) A (7DIN-DIN/REMOTE) @ £3.00 each  
 . . . . . LASERBUG Cassette Lead(s) B (7DIN/JACKS) @ £3.00 each  
 . . . . . x 20p to cover postage and packing

NAME . . . . . ADDRESS . . . . .  
 . . . . .

I enclose a cheque/PO number . . . . . for £ . . . . . p made payable to LASERBUG

**ALL ITEMS NORMALLY EX STOCK – RETURN OF POST SERVICE**

*(This order form supercedes all others)*

(For items out of stock please allow 28 days for delivery)