Issue 9 Feb'83

IN THIS ISSUE:

- Softreview
- Letters
- Contacts
- Computer conversions Part III

. . . . . and lots more!

2 editorial

Issue 9 of LASERBUG sees us growing at a tremendous rate beyond all our wildest dreams. When LASERBUG was started back in April '82 we no way expected LASERBUG ever to reach its current size. We started off life as a local user group covering just London and the South East and now have become a major international BBC User Group.

First thing on this months news items is hush-hush information about the Electron from an unnamed source. Apparently there will be two versions – a model A and B, both with 32k RAM. Official Acorn sources say it will cost about £150.00 but the general opinion is that a figure around £120/£130 is more likely. Acorn are just waiting for the ULA's to be delivered now and the launch date is promised as May although considering the BBC Project think nearer to November (only joking). It runs as you all know BBC Basic with a few alterations. For instance it only has 1 sound channel and the MODEs only range from 0 to 6. It has no joystick ports – just a cassette, UHF and mains. Other sources say an Electron User Port will be included not compatible with any known interface. The operating system is only 4 or 8k. A Welcome Pack tape has already be written for the system. The Electron can so we have been told read BBC cassettes – no official confirmation of this yet. Of course some programs will have to be altered. The keyboard is fairly similar to the BBC's one except that there are no function keys and the editing section is different. A special key is used like a shift to enable single key word entry. The upgrade to Model B is done by "bolt on" packs which fit onto the back. All this information is unofficial and may be subject to change. It does however, I am sure you will agree, sound interesting.

This brings me onto another point. The idea of LASERBUG always was and always will be to provide support to the BBC Micro and the BBC Micro only. However three other devices – (i) Acorn Atom with BBC Basic board, (ii) Torch and (iii) Acorn Electron use BBC Basic and are mostly compatible with the BBC Micro. It seems logical therefore to open up LASERBUGs doors to include all users of BBC Basic. Now obviously this is a big step and our primary interest would always be towards the BBC Micro. However it is hard to ignore these other people. We would therefore welcome your comments on us expanding to include all users of BBC Basic.

Apparently at the moment there are 3000 voice synthesis chips floating around. A few of them have found their way into certain privileged people's machines but are not as yet generally available. Hopefully by the time you read this you might stand a chance of getting them from your local dealer. Let us know . . .

When this Editorial was written the first episode of Making The Most Of The Micro had just been screened. Elsewhere in this magazine we let you know our opinions of it.

Details of two different graphics boards have come to light – one available now but the other soon to be developed. The first is from the Tangerine Users Group known as Video 80/82. It provides a display of either 256 x 256 or 512 x 256 with 40 x 25 or 80 x 25 text without using your micro's RAM!!! Details from 16 Iddlesleigh Road, Charminster, Bournemouth, Dorset, BH3 7JR. The other graphics board will eventually come from Acorn. You must know of the extra three zeros after a VDU19 statement – with the new graphics board you will be able to control each electron gun exactly i.e. a command saying 240 red, 12 blue and 0 green. Think of the possibilities this will give you.

Ever wanted to play around with control systems. Computer Accessories are producing a mains interface to allow your computer to have control over five 13 amp sockets to be turned on and off by your computer. Details from 69 Well Heads, Thornton, Bradford, BD13 3SJ.

A new BBC Buggy has been announced designed for use with the BBC Micro. The Buggy has three wheels and is created from a kit using just a screw-driver. The robot is about 6" square and driven by a 12 volt motor powered from the disk power supply socket. The robots capabilities include sensing obstructions either side by a special bumper, a light sensor and an infra red transceiver. Full software is included with the package which will cost around £140. Look out for it in episode 8 of the correct TV series. Incidentally there will be at least 1 more series of the computer program if not two!

Following on with the current craze, Acornsoft start a prize competition – £2000 in Acorn goods for the first person to solve the Castle of Riddles. Best of luck to everybody attempting it!

The 2nd Processors on the BBC Micro were shown working at the special BBC Computer exhibition during January 5th to 7th. All three devices were working fine. The two 8 bit devices will cost £195 and the 16 bit for £600, all prices without VAT.

Finally for this month a couple of LASERBUG matters. Although it is only February now please remember that as from April some members are due to renew their subscription. A reminder note will be sent to all members with their last magazine – after that we just will not send the magazines!

By the way, membership cards are coming very soon. If they are not with this issue then they will be by the following one. Please once you know it quote your membership number in all correspondence.

Please address all correspondence to:

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

It helps us considerably when sorting the mail if you can add one or two words in the top left hand corner of your envelope to describe what your letter is about. We regret that we cannot guarantee a reply to any letter unless you enclose a SAE.

Paul Barbour

oddspot

It's back to BASIC for the February Oddspot. Another graphical program based around the circle formula as usual. I won't tell you what it does but when it's finished the colours look like they are extra bright. The actual program was written by Trevor Lawford who receives £5 for his efforts.

```
L.
   10 REM (C) LASERBUG 1983
   20 MODE2
   30 VDU23;8202;0;0;0;29,640;512;
   40 FORX%=1TO360
   50    GCOL0,RND(6)
   60    MOVE0,0
   70    DRAWSINRAD(X%)*820,COSRAD(
X%)*820
   80    NEXT
   90 REPEATVDU19,RND(6),RND(6);0;
  100    FORP%=1TORND(10)+5:NEXT:UN
TILFALSE
```

competition 5

This months competition is very easy, or is it??? All you have to do is think up a competition for a future magazine along with the correct answer.

The competition should be original and imaginative but apart from that we will leave it up to you. The closing date is 15th March and winners will be notified as and when we use their idea. Prizes are as usual.

contents

SOUND PROGRAM: Soundplan
SUPPLIER: ME & P Micro Products, The Old Oast House, Malting Lane, Cambridge, CB3 9HF.
REQUIRES: 16k (?)
PRICE: £10.00
DESCRIPTION OF PROGRAM: If you want to explore into more detail how to use the ENVELOPE command and the advanced SOUND command then this program is unhesitantly for you. Making excellent use of the teletext display facilities this program allows you to manipulate all sound facilities of the computer and gives graphs of the current envelope. It allows for you to alter four envelopes at once with 9 different notes, the sound effects channel and a tune to try your experimentations out on. If I was asked by somebody how to really find out how to use the ENVELOPE command I would advise them without hesitation to buy this program.
PRESENTATION: ★★★
USEFULNESS: ★★★★
USE OF GRAPHICS: ★★★★
VALUE FOR MONEY: ★★★

–o0o–

ARCADE GAME PROGRAM: Arcadians
SUPPLIER: Acornsoft, 4a Market Hill, Cambridge, CB2 3NJ.
REQUIRES: 32k + 6522 VIA chip (joystick optional)
PRICE: £9.95
DESCRIPTION OF PROGRAM: Arcadians is the Acornsoft version of the arcade game Galaxians. For the newcomer to this type of game Arcadians is a kind of super space invaders game. As well as a normal kind of invader game every few seconds a group of aliens branch away from the main bunch to swoop down on you. Whilst they are doing this they are worth twice as many points. For those who find space invaders a bit boring you would certainly enjoy this. Personally this is the best game I have reviewed so far for the BBC Micro and as you would have seen from our top program list last month this one comes top without a doubt. Do send us your high scores on this game.
PRESENTATION: ★★★★★
ADDICTIVE QUALITY: ★★★★
USE OF GRAPHICS: ★★★★
VALUE FOR MONEY: ★★★

–o0o–

ADVENTURE PROGRAM: Adventure
SUPPLIER: Level 9 Computing, 229 Hughenden Road, High Wycombe, Bucks., HP13 5PG.
REQUIRES: 32k (16k version available with shorter screen messages)
PRICE: £9.90
DESCRIPTION OF PROGRAM: If I was asked to show an adventure program that was as much like the original ones as possible there are only a few programs that would come to mind – this one is very close to the top of the list. If you do not know about adventure programs then see our What Adventure? article in the last issue. If you do know about them and enjoy playing them then you would be a fool not to have this one. As well as playing the original game it has 70 extra rooms!
PRESENTATION: ★★★
SIZE: 27½k
COMPLEXITY: ★★★★
VALUE FOR MONEY: ★★★

–o0o–

ARCADE GAME PROGRAM: Brick 'Em In
SUPPLIER: Software for All, 72 North Street, Romford, Essex.
REQUIRES: 32k
PRICE: £8.00 (?)
DESCRIPTION OF PROGRAM: To describe this program it might sound like a version of Monsters but I should say from the beginning that it is really nothing like that. You are in an area with a monster (or many monsters, depending on how you choose to play) and a number of bricks. What you have to do is to build a kind of trap with these bricks with yourself as the bait. When the monster enters the trap you have to quickly get out of the trap and fill in the gap with another brick. Then you must brick in the monster completely. The quicker you do this the higher a score you get. A different game by all means and one I enjoyed.
PRESENTATION: ★★★
ADDICTIVE QUALITY: ★★★
USE OF GRAPHICS: ★★★★
VALUE FOR MONEY: ★★★

–o0o–

EDUCATIONAL PROGRAM: Castle
SUPPLIER: Swift Link Software, 118-120 Wardour Street, London, W1V 4BT.
REQUIRES: 32k
PRICE: £4.00 (?)
DESCRIPTION OF PROGRAM: This program is an informal way of testing very young children on some basic abilities in maths. The scenario is that you have to guide a man to a castle but every few moves you will be stopped and asked a question which you must get right to continue. The game gets progressively harder and will be enjoyed by children.
PRESENTATION: ★★★
FOR AGES: 6 to 8
SUBJECT: Basic maths
USEFULNESS: ★★★
NUMBER OF USERS: 1
VALUE FOR MONEY: ★★★

–o0o–

ARCADE GAME PROGRAM: Galactic Firebird
SUPPLIER: Kansas City Systems, Unit 3, Sutton Springs Wood, Chesterfield, S44 5XF.
REQUIRES: 32k (joystick optional)
PRICE: Around £10 (check actual price on 0246-850357)
DESCRIPTION OF PROGRAM: Kansas are a well-known name in TRS-80 software and have now started on the BBC Market with Galactic Firebird. If you take some of the features from space invaders, galaxians and defender then you might have an idea of what this game is like. It is really a very advanced kind of Space Invaders with some major differences. Some of the things you must shoot are pacman like aliens which require several shots, a kind of swarmer that moves in remarkable patterns and ICBM's that explode over a wide area. This game has so many different parts that it would be too hard to go into all of them. If it were not for two things I would say this game is equivalent to the Acornsoft standard: (i) the explosion of your ship is rather dull in comparison with the rest of the game and (ii) there is a bug which means that if the game isn't loaded in on a cold (i.e. memory completely clear by turning the computer off and then on again) system some colours can be altered making certain aliens impossible to see and others very odd colours. A good start from Kansas and I look forward to seeing their next offering.
PRESENTATION: ★★★
ADDICTIVE QUALITY: ★★★★
USE OF GRAPHICS: ★★★★
VALUE FOR MONEY: ★★★

–o0o–

GRAPHICS PROGRAM: Drawing
SUPPLIER: BBC Soft, 35 Marylebone High Street, London, W1M 4AA.
REQUIRES: 16k
PRICE: £10.00
DESCRIPTION OF PROGRAM: This is one of a few graphics program available for the BBC Micro. The list of features of this program is impressive – polygon, sphere and cone drawing, grids, horizons. There is unfortunately something lacking with the program – a complaint I have made all too often with these BBC Soft programs. One of the most annoying things is the cursor control. This is done by pressing the direction you want to go in and then the number of positions. This is a rather cumbersome method which makes the program far less attractive than it ought to be. Personally I prefer the BBC Artist program reviewed in a previous magazine. This program could do with re-writing in parts to make it worth £10.
PRESENTATION: ★★★
USE OF GRAPHICS: ★★★
USEFULNESS: ★★
VALUE FOR MONEY: ★★

–o0o–

ARCADE GAME PROGRAM: Atlantis
SUPPLIER: IJK Software, 9 King Street, Blackpool, Lancs. *(note the new address)*
REQUIRES: 32k
PRICE: £6.95
DESCRIPTION OF PROGRAM: This is a version of the arcade game Scramble. The idea is to move along a mountainous terrain whilst destroying all alien craft in sight. The game is very colourful and in itself plays quite a good game but the unevenness of the scrolling shows the program up against its Acornsoft counterparts. Not bad for £7 but it might be worth splashing out another £3 and get the Acornsoft version.
PRESENTATION: ★★★
ADDICTIVE QUALITY: ★★★
USE OF GRAPHICS: ★★★★
VALUE FOR MONEY: ★★★

–o0o–

BUSINESS PROGRAM: Mailing_A
SUPPLIER: Micro Aid, 25 Fore Street, Praze-an-Beeble, Camborne, Cornwall, TR14 0JX.

REQUIRES: 16k
PRICE: £3.95
DESCRIPTION OF PROGRAM: This program tries to form a cheap way of holding a mailing list. However, the way it tries to do this is its downfall. The BASIC program is fine **BUT** the data for the list is held in data statements at the end of the program. This makes the program hard to use and unflexible. If it was re-written to hold the information in the memory then it would be worth something. Without this it is pointless.
PRESENTATION: ★★★
MAXIMUM SIZE: Above 100 names and addresses
FLEXIBILITY: ★★
EASE OF USE: ★★
VALUE FOR MONEY: ★★

–o0o–

ARCADE GAME PROGRAM: Frenzy
SUPPLIER: Persoft, Freepost, Baildon, Shipley, W. Yorkshire,BD17 5BR.
REQUIRES: 32k (not suitable for a Model A as the cassette card says)
PRICE: £5.75
DESCRIPTION OF PROGRAM: The idea of this game reminds me of before I passed my driving test. You drive around an alien city and get points by knocking down the local population, traffic lights, fire hydrants and dogs – all without a brake in your landspeeder. The problem is each time you kill something an "anti-matter" block appears on the screen which will kill you if you touch it. The game sounds very good but doesn't somehow quite live up to expectations.
PRESENTATION: ★★★
ADDICTIVE QUALITY: ★★★
USE OF GRAPHICS: ★★★
VALUE FOR MONEY: ★★★

–o0o–

LASERBUG would like to thank ME & P Micro Products, Acornsoft, Level 9 Computing, Software for All, Swift Link Software, Kansas City Systems, BBC Publications, IJK Software, Micro Aid and Persoft for supplying us with review material.

As a quick note would companies supplying us with review software in future make sure that you say clearly how much the program costs.

wallball

Wallball is a classic breakthrough type game for people with games paddles (otherwise known as joysticks – Ed.). It is unusual in that the cat can be moved in all directions, and the way you are moving when you hit the ball affects the direction it bounces off, so that the game feels much more like a real bat and ball game.

Once you have mastered the way the paddle controls the bat, the game becomes an interesting challenge, and the two statements REMed out at the end of PROCPLAY can be inserted to give the walls the ability to fight back! Each set of walls gives a score of 1000 in total, with the further walls scoring more than the nearer ones. The skill is in getting the ball trapped between the walls, as in all the best breakthrough games.

To start the ball, press the fire button on the paddles.

```
L.
    10REM            WALLBALL
    20REM       by Peter Voke
    30:
    40REM          January'83
    50:
    60REM         Version 1.0
    70:
    80REM Takes up ~3.17k memory
    90REM (6.67k with variables)
   100:
   110REM     Uses MODEs 2 & 4
   120:
   130REM        Requires 32k
   140:
   150REM    Written on OS 0.1
   160:
   170REM    (c) LASERBUG 1983
   180:
   190    :::::
   200:
   240 PROCSET
   250 MODE4: PROCSPEED
   260 MODE2: PROCOURT
   270 FOR Ball%=1 TO 5: PROCPLAY:
NEXT
   280 PROCMORE
   290
   300END
   310
   320
   330:::::::::::::::::::::::::::::::::::
::::::
   340
   350DEFPROCSET
   360
   370   *FX9,0
   380   ENVELOPE1,129,-2,8,0,10,40,
0,127,0,-20,-30,100,100
   390   VDU23,224,12,12,12,12,12,12
,12,12
   400   VDU23,225,15,15,15,15,15,15
,15,15
   410   VDU23,226,2,5,5,5,2,0,0,0
   420   ON ERROR GOTO 280
   430
   440ENDPROC
   450
   460
   470:::::::::::::::::::::::::::::::::::
::::::
   480
   490DEFPROCSPEED
   500
   510   CLS: *FX15,0
   520   INPUT TAB(5,10) "MAXIMUM SP
EED?"''SPC5"Enter a number, 1 to 1
0,"''SPC5"and Press RETURN : " M%
   530   IF M%<1 OR M%>10 THEN 510
   540   M%=2*M%-1
   550
   560   S%=0: A%=1000: 0%=&404
   570   K%=-1: L%=0
   580
   590ENDPROC
   600
   610
   620:::::::::::::::::::::::::::::::::::
::::::
   630
   640DEFPROCOURT
   650
   660   VDU4,19,4,5;0;19,5,6;0;19,6
,4;0;
   670   COLOUR 11
   680   PRINT TAB(6,30) "WALLBALL"
   690   COLOUR15
   700   PRINT TAB(0,2) "ball" SPC5
```

continued on page 18

competition 4 results

How did you get on with producing a program to give all the possible permutations of the word CHRISTMAS? Well, the winning program appears below:

```
L.
    10 MODE1
    20 DIMA$(9)
    30 FORI%=1TO9
    40   READA$(I%)
    50   NEXT
    60 FORA%=1TO9
    70   FORB%=1TO9:IFB%=A% THEN270
    80     FORC%=1TO9:IFC%=A% ORC%=
B% THEN260
    90       FORD%=1TO9:IFD%=A% ORD
%=B% ORD%=C% THEN250
   100         FORE%=1TO9:IFE%=A% O
RE%=B% ORE%=C% ORE%=D% THEN240
   110           FORF%=1TO9:IFF%=A%
 ORF%=B% ORF%=C% ORF%=D% ORF%=E% T
HEN190
   120             FORG%=1TO9:IFG%=
A% ORG%=B% ORG%=C% ORG%=D% ORG%=E%
 ORG%=F% THEN180
   130               FORH%=1TO9:IFH
%=A% ORH%=B% ORH%=C% ORH%=D% ORH%=
E% ORH%=F% ORH%=G% THEN170
   140                 FORJ%=1TO9:I
FJ%=A% ORJ%=B% ORJ%=C% ORJ%=D% ORJ
%=E% ORJ%=F% ORJ%=G% ORJ%=H% THEN1
60
   150                   PRINTA$(A%
);A$(B%);A$(C%);A$(D%);A$(E%);A$(F
%);A$(G%);A$(H%);A$(J%)
   160                   NEXT
   170                 NEXT
   180               NEXT
   190             NEXT
   200           COLOUR2:PRINT'"PRE
SS ANY KEY TO CONTINUE"
   210           Q=GET
   220           COLOUR3
   230           PRINT
   240           NEXT
   250         NEXT
   260       NEXT
   270     NEXT
   280   NEXT
   290 END
   300 DATAC,H,R,I,S,T,M,A,S
```

and it was written by Dr. Patrick Smith of Upminster, Essex who has his membership increased until Issue 16. Well done Dr. Smith.

computer conversions pt. III

NEXT: Standard BASIC. Most versions do require you to state the variable.
NOT: Fairly standard BASIC which gives result of an operation is reversed. Used mainly in IF statements.
NULL: Unusual word. Allows you to set the number of nulls after each line.
OLD: Unusual word. Used to recover a NEWed program.
ON/GOTO or **GOSUB:** Mostly standard BASIC although not available on ZX or Atom. Only ON/GOTO is available on the UK101.
ONERROR: Fairly unusual BASIC. Sometimes it must be ONERRORGOTO and on the Atari 400/800 is known as TRAP.
OPEN: Mostly standard BASIC although sometimes has different names and syntax. Known on the BBC Micro as OPENIN (for input) or OPENOUT.
OPT: Unique to BBC Micro. Used in assembler to set listing and error events.
OUT: Fairly unusual BASIC. Used to send a single byte to the specified port. On the BBC Micro this can be done to printer at least with VDU1.
OR: Mostly standard BASIC.
PAGE: Unique to BBC Micro. A pseudo-variable giving the memory location where the BASIC program starts. Has to be found out using PEEK and USR commands on other micros and set using POKEs. Standard BASIC (except for T1 99/4a).
PEEK: Examines the contents of a single byte of memory. Replaced by ? on the BBC Micro. The locations used on one micro are not transferable to another.
PI: Fairly standard BASIC. Holds the value of π. On the BBC Micro is given the value of 3.14159265.
PLOT: Unique to BBC Micro. It is a multi-purpose graphics command covering point, line and triangle drawing.
POINT: Unique to BBC Micro. Returns the colour at any point on the screen.
POKE: Standard BASIC (except TI 99/4a). Alters the value of a single memory location. Replaced by ? on the BBC Micro. Again memory locations are not transferable between machines.
POS: Unique to BBC Micro. Gives X position of cursor. Available on other machines via a different statement or by PEEKs.
PRINT: Standard BASIC.
PROC: Unique to BBC BASIC. A kind of subroutine.
RAD: Unique to BBC BASIC. Converts degrees to radians.
RANDOMIZE: Fairly unusual BASIC. Sets the random number generator. Use RND with a negative argument on the BBC Micro.
READ: Fairly standard BASIC (except for Atom and ZX81). READs some information from a DATA statement.
REM: Standard BASIC (except for UK101). Gives a REMark on the program.
RENUMBER: Unique to BBC Micro. RENUMBERs all the lines. Found on other machines only in toolkit programs.
REPEAT: Unique to BBC Micro. Part of the REPEAT UNTIL loop.
REPORT: Unique to BBC Micro. Gives last error message.
RESTORE: Fairly standard BASIC. Resets the DATA pointer.
RESUME: Fairly unusual BASIC. Takes machine back to the line where an error occured. Could be replaced with GOTOERL on BBC Micro.
RETURN: Standard BASIC. RETURNs you to the statement following the last GOSUB.
RIGHT$: Standard BASIC. Returns the specified number of characters from the right of the main string.
RND: Standard BASIC. Generates a random number although the range of number produced differs from micro to micro sometimes.
SGN: Standard BASIC.
SIN: Standard BASIC.
SOUND: Unique to BBC Micro with same syntax. Although several computers have sound capabilities some use a different command (e.g. Beep).
SPC: Fairly standard BASIC. Returns the specified number of spaces. Can only be used in PRINT or INPUT on BBC Micro.
SQR: Standard BASIC.
STEP: Fairly standard BASIC. Adds a step factor to FOR loops.
STOP: Fairly standard BASIC. On some machines this replaces END. On others like the BBC Micro it prints out STOP AT LINE ////. On some STOP is used so it can be restarted using CONT.
STR$: Standard BASIC.
STRING$: Fairly unusual BASIC. Returns a string containing n times of its string argument.
SYSTEM: Standard BASIC on machines where it is required. Resets computer to default but NOT like BREAK. E.g. on the MZ-80A you have to load in the BASIC from tape or disk. BYE (the Sharp's equivalent of SYSTEM) clears the BASIC.

NEXT MONTH: **T to Z** *(if we can find a BASIC word beginning with Z!).*

Paul Barbour

continued from page 4

```
"score"
   710   VDU5
   720
   730   Wall%=1248
   740   FOR I%=5 TO 1 STEP -1
   750     Wall%=Wall%-96: PROCWALL(I
%)
```

continued on page 7

6 assembler prog – BBC micro pt II

When writing programs in machine code the most important thing to remember is to think clearly and carefully. Before you put anything into code think out exactly what you want to do, first in general terms then in more detail, until you have broken your problem down into a series of very simple stages *(try doing a flowchart – Ed.)*. 8 bit microprocessors can only work in very short steps so you will find even simple programs need to be broken down into many stages. Imagine writing in BASIC with only one variable and one arrary: 6502 programs can be no more complicted.

Manipulation of data by the processor is done by instructions which work on the accumulator. This can be thought of as a simple variable but it will only hold one byte of information i.e. a number in the range 0 to 255. Exactly how this byte is used will be dependent on the programmer, for example it may be the character code from a character string or part of an integer. It would be helpful if you could see what happens to the accumulator when you have performed your machine code routine. Luckily BBC BASIC allows this with the USR function. This is like any other integer function in as much as it returns a number as its value dependent on the number passed to it but the argument given to the USR function tells it where to find a machine code routine and the value returned will be dependent on the value of the accumulator and other registers when it is finished. To see what has happened in your routine try using

```
>PRINT~USR(code%)
```

This will show the registers in hexadecimal. Many people find this number system confusing and would rather use decimal. However I would advise you to become familiar with it since it is much closer to the way the machine thinks. If you are not familiar with the hex try the following:

```
>L.
    10 FORnum%=0TO1000
    20    PRINTnum%,~num%
    30    wait=INKEY(50)
    40    NEXT
```

This will print out the numbers one at a time in decimal and hexadecimal. You will notice they start off the same but 10 decimal is represented by "A", 10 in hex being 16 in decimal. I said earlier that a byte can hold a value between 0 and 255. You will notice that this is 0 to FF in hexadecimal – this is no coincidence. Two hex digits can exactly represent one byte. Any number printed in hex is therefore split into its bytes for inspection. This is why we want to use it here.

Set up one of the user definable keys as such:

```
>*KEY0"PRINT~USR(code%)|M"
```

then try running the following assembler program to create some machine code for it:

```
>L.
    10 DIMP% 50
    20 [.code%
    30 LDA#7
    40 RTS
    50 ]
>RUN
0E2C                .code%
0E2C A9 07          LDA#7
0E2E 60             RTS
>PRINT~USR(code%)
   31000007
>
```

Each time you press the user defined key after you have run the program you will find a number printed ending in 07. Now run a modified program:

```
>L.
    10 DIMP% 50
    20 [.code%
    30 LDA#2
    40 RTS
    50 ]
>RUN
0E2C                .code%
0E2C A9 02          LDA#2
0E2E 60             RTS
>PRINT~USR(code%)
   31000002
>
```

This time the number printed will end in 02. Each time you call the USR function it jumps to the subroutine that you have created, executes it and then returns a value when it reaches the RTS instruction. The machine code has an action which is suggested by the mnemonics in the assembler. "LDA" is short for LoaD Accumulator. When the processor encounters this instruction it needs to know what it has to load as well as what to do. This is called the operand and in this case is#2.#stands for number so LDA#2 means load the accumulator with the number 2. When the return instruction is encountered the accumulator will therefore contain 2 so thus is returned by the USR function. Modify the program again:

```
L.
    10 DIMP% 50
    20 [.code%
    30 LDA#RND(255)
    40 RTS
    50 ]
>RUN
0E31                .code%
0E31 A9 10          LDA#RND(255)
0E33 60             RTS
>PRINT~USR(code%)
   31000010
>PRINT~USR(code%)
   31000010
>RUN
0E31                .code%
0E31 A9 81          LDA#RND(255)
0E33 60             RTS
>PRINT~USR(code%)
   B1000081
>PRINT~USR(code%)
   B1000081
>RUN
0E31                .code%
0E31 A9 04          LDA#RND(255)
0E33 60             RTS
>PRINT~USR(code%)
   31000004
>RUN
0E31                .code%
0E31 A9 4F          LDA#RND(255)
0E33 60             RTS
>PRINT~USR(code%)
   3100004F
>
```

Run it a few times, each time pressing the function key (defined earlier) a few times. You will notice that each time you run the program the print will give a different result. This is because each time you assemble it you include a different random number. On the other hand each time you do a PRINT after a run you will get the same result. This is because the random number you started with will be built into the machine code by the assembler.

Now try this:

```
>L.
    10 DIMP% 50
    20 [.code%
    30 LDA&70
    40 RTS
    50 ]
>RUN
0E2D                .code%
0E2D A5 70          LDA&70
```

```
0E2F 60          RTS
>PRINT~USR(code%)
   33000000
```

The operand for the load instruction has now lost the## The accumulator will now be loaded **from a memory location** in this case hexadecimal 70.

To show the define another key thus:

```
>*KEY1"?&70=RND(255):PRINT~USR(co
de%)!M"
```

Each time you see this you will find a different number is printed. Replace the RND with a number and you will find this appearing. What is happening is that in BASIC you are putting a value into the byte at hexadecimal location 70 and then you are reading it back in machine code. Not very exciting but it's a start.

Finally, to give you a clue why you may want to do all this try the program below:

```
L.
    10 DIMP% 50
    20 [.code%
    30 LDA&70
    40 JSR&FFEE
    50 RTS
    60 ]
    70 REPEAT
    80    ?&70=RND(224)+31
    90    CALLcode%
   100    UNTILFALSE
```

Can you see why this is equivalent to the pure BASIC line REPEAT:VDURND(224)+31:UNTIL FALSE ???

Nick Goodwin

computer prog. review 1

From this month onwards for 10 months we are going to review a single episode of the Computer Programme II. We could have reviewed the whole series briefly in one article but on consideration we thought that the program is worth more than that. Hence we will go into depth about a one episode each time. Obviously the reviews will go till well after the series has finished but I am sure you will bear with us. Your comments on the programmes are always welcome and any suitable letters we receive on the topic will be printed.

The Computer Programme II – Making The Most Of The Micro
Episode 1 – The Versatile Machine
Presented by Ian McNaught Davies with Richard Gomm and John Coll
Produced by David Allen

The first episode starts with Ian McNaught Davies playing Acornsoft's Monsters. His opening speech leads us into the plans and aims of the programme. "Well, until recently it's been fashionable to say that the microcomputer's coming on the market were only really for playing games and that most people never find serious uses for them. Well, over the next 10 programmes we'll be seeing just what can be done with micro's and looking at the principles of how you can use them yourself. That's why we're calling this series Making The Most Of The Micro." Hopefully everyone reading LASERBUG knows that the BBC Computer can be used much, much more than just playing games (although 95% of the time spent on all home computers is to do with games!)

Next we see Richard Gomm who is disabled with cerebral palsy. With cerebral palsy you can't walk and your limb movements are unpredictable but head movement is controllable. At the moment he is studying for a PhD in philosophy at Swansea. Some friends of his at York University suggested to him in 1979 that he might use computers to help him write as using a typewriter for Richard was very hard. Fortunately for Richard this was a time when cheap computers were becoming available.

With no experience in computing he managed to get a computer system and a commercially available word-processing package which opened his eyes to this new potential. Now he was able to get his letters just right before printing them out on his printer. He operated the computer by a styli attached to his head which you can see he has now become well accustomed to and proficient at using.

However Richard soon realised that word processing was just the start and he was underusing the computers potential. Learning everything from books and manuals he soon wrote his own word processing package with special features that most people would not need.

He now uses a VIC 20 and some hardware to control up to 6 mains devices (lamp, tape recorder, radio, printer, TV) via the keyboard or a pressure pad which gave him a new freedom not possible before.

The advent of the computer you can see has certainly opened up a new world for Richard and I am sure many other disabled people. Richard Gomm sets an example to us all.

Next we move back into the studio and Ian McNaught Davies is joined by John Coll. They start off with the basic (no pun intended) set up for the BBC Micro – computer, display system (RGB Monitor) and tape recorder. They then load in a program which takes 50 seconds to finish (not 20 or 30 as Ian McNaught Davies says!) which shows how the cassette system works.

A single floppy disk drive is then introduced into the system (note that all of the disks used in this episode are only the single ones which would infer that out of the many devices which are ready i.e. the speech synthesis chip is heard the dual floppy disk is still no where near ready???) John Coll quotes the interesting figures that disks work 100 times as fast but cost 7 times as much as tape.

Finally an Epson MX80 printer is brought in. The VIEW chip was apparently already fitted inside the machine. With the two things they load the script of the program into the computer from disk and start printing it out on the Epson.

The fourth scene is at a sail making factory. The design of aerodynamic sails requires a great deal of complicated calculations and to do this the company has used a computer since the late fifties. Very recently they have updated their system to an 80 column PET with dual discs and printer.

We though get to take a look around the 20 year old computer. The information was entered on paper tape and put into the computer. The computer then gave out the measurements on another paper tape which was then put onto a teletype to be read. One interesting meter on the computer was a "Percentage Program Efficiency" reading???

Anyway, we then was shown inside the actual beast. The backing store consisted of a large magnetic drum rotating at 6000 rpm which is replaced in their new system by floppy disks.

The main part of the computer consisted of a giant "rack" taking up the majority of the space. In the rack were many logic gates, each one the size of a man's hand. Ian McNaught Davies then showed the principles of AND and OR gates. Then we were shown what had replaced this huge valve machine a single chip with several thousand of those logic gates on it. About 12 of those chips with about 36 000 logic gates would take up the same room as 1 logic gate in the old computer.

Now Ian McNaught Davies moves inside the BBC Computer and sits down quite happily on the CPU (this must have been television trickery or he wouldn't be sitting happily on the CPU when you consider how hot it gets!!!). From his viewpoint he looks at several areas of the computer and demonstrates with the aid of the micro how the programs from tape go into the memory and the difference between RAM and ROM.

To end the programme we are shown the BBC Computer doing several different things. Firstly a graphics program which shows a 3 dimensional house being rotated. It is then stopped and the screen picture drawn out on an X-Y plotter.

Next a sound recognition system is shown with a graphical representation being given of his speech as well as a cartoon man opening and closing his mouth in time with the presenters.

Finally we see a pressure pad connected to the BBC Micro with a graphical representation and numeric of the weight applied to it (from which we find out that Ian McNaught Davies weighs 205 lbs!). Pressing another button brought the speech chip into action which spoke the weight on the screen.

The final part of the program says as much as the first. "In later programmes we'll look in more detail at how we can use it for displaying graphics, making music, controlling equipment and handling words. Well, amongst many other things we'll be talking to the computer and getting it to talk back to us and looking at how it can monitor what's happening in the world around it."

Overall this first episode was interesting but didn't really show a great deal in facts. It does though provide a very good introduction to the series and shows you a little of what is to come. A good start to the new series.

Next Month: Getting Down To BASIC

Paul Barbour

continued from page 4

```
760   NEXT
770
780   GCOL0,6
790   MOVE0,900: DRAW1279,900
800   DRAW1279,95: DRAW0,95
810   PROCSCOR
820
830ENDPROC
```

continued on page 18

8 seasonal 5 – dating

Certain months are harder than others to think up seasonal programs. The only notable event in February to write a program for was St. Valentine's Day. Hence I wrote the program below based around computer dating.

The idea of the program is to test the compatibility between you and your potential partner. Both of you are asked 30 different questions from which the computer will determine the percentage compatibility between the two of you.

The results should not be taken too seriously . . .

```
L.
   10 REM Seasonal 5 - February
   20 REM for St.Valentine's Day
   30 :
   40 REM      Computer Dating
   50 REM      by Paul Barbour
   60 :
   70 REM           18/1/83
   80 :
   90 REM        Version 1.0
  100 :
  110 REM Takes up ~4.01k memory
  120 REM (8.15k with variables)
  130 :
  140 REM       Uses MODE 7
  150 :
  160 REM     Suitable for 16k
  170 :
  180 REM    (c) LASERBUG 1983
  190 :
  200     :::::
  210 :
  220 MODE7
  230 VDU23;8202;0;0;0;
  240 PROCteletext
  250 PROCtitle
  260 PROCvariables
  270 PROChim
  280 PROCquestions
  290 PROCher
  300 PROCquestions
  310 PROCProcess_data
  320 PROCresults
  330 END
  340 :
  350     :::::
  360 :
  370 DEFPROCteletext
  380 dble$=CHR$141
  390 whtbgrnd$=" "+CHR$157
  400 blue$=CHR$132
  410 grn$=CHR$130
  420 ylw$=CHR$131
  430 red$=CHR$129
  440 ENDPROC
  450 :
  460 DEFPROCtitle
  470 PRINTTAB(11,8);dble$;"COMPUT
ER DATING:"
  480 PRINTTAB(11);dble$;"COMPUTER
 DATING:"
  490 PRINTTAB(10,11);dble$;"Suita
bility Tester"
  500 PRINTTAB(10);dble$;"Suitabil
ity Tester
  510 PRINTTAB(0,20);'"NOTE: This p
rogram is meant to be taken"''"ligh
tly and no serious conclusions"''"s
hould be drawn from the results."
  520 null%=INKEY(1000)
  530 ENDPROC
  540 :
  550 DEFPROCvariables
  560 DIMdata%(1,29)
  570 DIMname$(1)
  580 rating%=0
  590 ENDPROC
  600 :
  610 DEFPROChim
  620 CLS
  630 PRINTwhtbgrnd$;dble$;grn$;"
 Questions for";blue$;"Him";grn$;"
to answer..."
  640 PRINTwhtbgrnd$;dble$;grn$;"
 Questions for";blue$;"Him";grn$;"
to answer..."
  650 VDU28,0,24,39,2
  660 sex%=0
  670 ENDPROC
  680 :
  690 DEFPROCquestions
  700 CLS
  710 PRINTylw$;"      Firstly,  wh
at is your name:"
  720 PRINT'ylw$;
  730 INPUTname$(sex%)
  740 IFLEN(name$(sex%))>=10PRINT'
ylw$;"That's a bit too long - have
n't you got";ylw$;"a nickname or s
omething to make it"'ylw$;"more fr
iendly ???  Try something else:":G
OTO720
  750 CLS
  760 PRINT'ylw$;"OK ";name$(sex%)
;",  now I am going to give"'ylw$;
"you 30 different questions to ans
wer"
  770 PRINTylw$;"to see what I can
 find out about you."'ylw$;"When I
 have the data from both you and"
  780 PRINTylw$;"your Potential Pa
rtner I will give you"'ylw$;"a Per
centage suitability rating."
  790 PRINT'ylw$;"Press the";grn$;
"RETURN";ylw$;"key to continue..."
  800 REPEATUNTILGET=13
  810 CLS
  820 PRINT'ylw$;"      The first t
en questions are about"'ylw$;"how
you see yourself.  To every"
  830 PRINTylw$;"question here ans
wer either Yes (by"'ylw$;"Pressing
 Y) or No (by Pressing N)."
  840 PRINT'ylw$;"Press the";grn$;
```

```
"RETURN";ylw$;"key when ready..."
  850 RESTORE950
  860 REPEATUNTILGET=13
  870 FORquestion%=0TO9
  880    CLS
  890    READatribute$
  900    PRINT'ylw$;"Do you conside
r yourself to be:"'ylw$;atribute$
  910    PRINT'ylw$;"Press";grn$;"Y
";ylw$;"or";grn$;"N"
  920    REPEATanswer$=GET$:UNTILan
swer$="Y"ORanswer$="N"
  930    IFanswer$="Y"THENdata%(sex
%,question%)=1:ELSEdata%(sex%,ques
tion%)=0
  940    NEXT
  950 DATAShy,Extrovert,Adventurou
s,Family type,Clothes-conscious
  960 DATAGenerous,Outdoor type,Cr
eative,Practical,Interllectual
  970 CLS
  980 PRINT'ylw$;"      The other 2
0 questions are about"'ylw$;"your
likes and dislikes.  To each"
  990 PRINTylw$;"question answer e
ither L for like,  D"'ylw$;"for di
slike or I for indifferent."
 1000 PRINT'ylw$;"Press the";grn$;
"RETURN";ylw$;"key when ready..."
 1010 REPEATUNTILGET=13
 1020 RESTORE1150
 1030 FORquestion%=10TO29
 1040    CLS
 1050    READintrest$
 1060    PRINT'ylw$;"What do you th
ink of:"'ylw$;intrest$
 1070    PRINT'ylw$;"Press";grn$;"L
";ylw$;"if you like it"
 1080    PRINTylw$;"Press";grn$;"D"
;ylw$;"if you dislike it"
 1090    PRINTylw$;"Press";grn$;"I"
;ylw$;"if you are indifferent abou
t it"
 1100    REPEATanswer$=GET$:UNTILan
swer$="L"ORanswer$="D"ORanswer$="I
"
 1110    IFanswer$="L"THENdata%(sex
%,question%)=1
 1120    IFanswer$="D"THENdata%(sex
%,question%)=-1
 1130    IFanswer$="I"THENdata%(sex
%,question%)=0
 1140    NEXT
 1150 DATAPoP music,Fashion,Pubs a
nd Clubs,Sport,Pets
 1160 DATAFolk music,Jazz,Travelli
ng,Cinema,Good food
 1170 DATAPolitics,Classical music
,Art/Literature,Live theatre,Scien
ce/Technology
```



```
 1180 DATACreative writing/Paintin
g,Poetry,Philosophy/Psychology/Soc
iology,History/Arcaeology,Conversa
tion
 1190 ENDPROC
 1200 :
 1210 DEFPROCher
 1220 CLS
 1230 VDU7,26
 1240 PRINTwhtbgrndg$;dble$;grn$;"
 Questions for";red$;"Her";grn$;"t
o answer..."
 1250 PRINTwhtbgrndg$;dble$;grn$;"
 Questions for";red$;"Her";grn$;"t
o answer..."
 1260 VDU28,0,24,39,2
 1270 sex%=1
 1280 ENDPROC
 1290 :
 1300 DEFPROCprocess_data
 1310 CLS
 1320 VDU26,7
 1330 FORquestion%=0TO9
 1340    IFdata%(0,question%)=data%
(1,question%)THENrating%=rating%+1
 1350    NEXT
 1360 FORquestion%=10TO29
 1370    IFdata%(0,question%)=data%
(1,question%)THENrating%=rating%+2
:GOTO1390
```

continued on page 21

10 contacts

If you would like to get in touch with other local users but do not want to get involved in the organisations of a local group, try looking in the list below for some contacts. If you would like to be put on this list, please write to us at the usual LASERBUG address and mark the envelope Contacts.

- Dr D E Susans
  19 Rushout Avenue,
  Harrow,
  Middx. HA3 0AS
  01-907 1964
- Wayne Wealleans
  7 Hillman's Cottages,
  Ongar Road,
  Abridge,
  Essex RM4 1UL
  Theydon Bois 4154
- Dr John Willis
  27 Park Road
  Hampton Hill,
  Middx. TW12 1HG
  01-979 6655
- John Matchett
  01-940 9361
- W G Morley
  107 Sandfield Road,
  Arnold,
  Nottingham NG5 6QF
  0602-267635
- Mr John F Murphy
  10 Birchmore,
  Brookside,
  Telford,
  Shropshire
  0952-595959
- Mr P S Murphy
  01-740 8082
- G Musgrove
  16 Orchard Road, South Croydon
  Surrey CR2 9LU
  01-651 0011
- Nicholas D Lamb
  23 Gaywood Close,
  Caister-on-Sea,
  Gt. Yarmouth,
  Norfolk NR30 5RD
  0493-728442
- N Lambert
  11 Vinson Close,
  Orpington,
  Kent BR6 0EQ
- R W Macmillan
  38 Box Ridge Ave.,
  Purley,
  Surrey CR2 3AQ
  01-660 5615
- Mr Mark Cook
  165 Witham Road,
  Black Notley,
  Braintree,
  Essex CM7 8NB
  0376-23084
- R Hall
  89 Hicks Ave.,
  Greenford,
  Middx UB6 8EZ
  01-578 9136
- David N Hardwick
  Hagley 885183
- Laurie Hartman
  178 East Barnet Road,
  Barnet,
  Harts EN4 8RD
- John Harvey
  93 Southfield,
  Hessle,
  N. Humberside HU13 0ET
  0482 645905
- A R J Hunt
  22 Meyer Road,
  Erith,
  Kent
  Erith 38620
- David H M Glew
  20 Barnfield Wood Road,
  Beckenham,
  Kent
  01-650 1365
- E E Godfrey
  6 Wharf Road,
  Wraysbury,
  Nr. Staines,
  Middx. TW19 5JQ
  Wraysbury 2624
- D T Green
  Bay House, Painter's Forstal,
  Faversham,
  Kent ME13 0D
  Faversham 6483
- Peter Greenall
  77 Memorial Ave.,
  Stratford,
  London E15
- Paul E Firman
  63 Lady Lane,
  Chelmsford,
  Essex CM2 0TQ
  0245 50432
- P Y Norman
  Flat 2,
  3 Cedar Gardens,
  Sutton,
  Surrey SM2 5EQ
  01-643 1944
- Mr J D Osborn
  8 Chytroose Close,
  Helston,
  Cornwall TR13 8UY
  Helston 4121 Ext. 7378
- James Stewart
  Lammas Field,
  Baring Road,
  Cowes,
  Isle of Wight.
  Cowes 292107
- Mr Dilbagh Singh
  29 Wingfield Mount,
  Bradford,
  West Yorkshire BD3 0AG
- Mr M F Shirt
  30 Orchard Road,
  West Northenden,
  Manchester M22 4FD
- E A Watson
  19 Darrell Close,
  Chelmsford,
  Essex CM1 4EL
  Chelmsford 83790
- Ian Dennison
  38 Tankerville Road,
  London SW16 5LP
  01-764 4958
- Mr R W Thompson
  Redhill 63232
- Stephen E Parsons
  16 Primrose Hill,
  Haverhill,
  Suffolk CB9 9LS
  0440-70422 Ext. 52
- D I Pearce
  Little Orchard,
  Brentmoor Road,
  West End,
  Woking,
  Surrey
- Brian D Piper
  73 Fairlawn Drive,
  East Grinstead,
  West Sussex RH19 1NS
  0273-23373
- P W Rautenbach
  25 Bishopsfield,
  Harlow,
  Essex CM18 6UJ
- Mr H E Robson
  Church Farm,
  Stewkley,
  Nr. Leighton Buzzard
  0525-24244

bu Paul Shaddick
325 Wickham Lane,
Abbey Wood,
London SE2 0NT
01-854 7766

- J M Stone
  8 Sandra Court,
  Spencer Road,
  Chiswick,
  London W4
  01-995 6144
- V A Amesbury
  33 King Edgar Close,
  Ely,
  Cambs. CB6 1DP
  Ely 3274
- C G Bartholomew
  102 Cranley Gardens,
  Muswell Hill,
  London N10 3AH
  01-883 2624
- Mr A J Beer
  2 Waterworks Cottages
  Dotton,
  Sidmouth,
  Devon EX10 0JY
- Rick Branston
  37 Princes Road,
  Romford,
  Essex
  Romford 61336
- Daniel O'Brien
  5 Highfields Mead,
  East Hanningfield,
  Chelmsford,
  Essex CM3 5XA
- B Buckland
  Purton 771669
- Simon Cheshire
  1 Knowsley Way,
  Hildenborough,
  Kent TN11 9LG
  Hildenborough 833108
- Dave Clare
  "Providence House",
  222 Townfields Road,
  Winsford,
  Ches. CW7 4AX
  Winsford 51374
- Dr P G Clayton
  Chigwell School,
  High Road,
  Chigwell,
  Essex
- Nicholas Clifton
  4 Bonchester Close,
  Chislehurst,
  Kent BR7 5FS
  01-467 5396
- R F Colin
  13 Pembroke Road,
  Moor Park,
  North Wood,
  Middx. HA6 2HP
- Peter M Corrie
  58 High Street,
  Haslemere,
  Surrey GU27 2LA
  0428 3838
- G Cox
  78 Napier Road,
  Gillingham,
  Kent
  0634 55475
- David Dade
  30 Hilden Park Road,
  Hildenborough,
  Tonbridge,
  Kent TN11 9BL
  0732 838603
- P J Davies
  'Treetops',
  Burnham Road,
  Althorne,
  Chelmsford,
  Essex CM3 6DP
  Maldon 740084
- Mr Chris Drage
  28 Ingersoll Road,
  Shepherds Bush,
  London W12.
- R.M. Timothy
  35 Potash Road,
  Billericay,
  Essex CM11 1DL
- Catherine Brown
  2 Plas Newydd,
  Southend-on-Sea,
  Essex, SS1 3AG.
  0702-587066.
- J R Dyer 0954-81074.
- Roger Gibbons,
  Lynford Recreation Road,
  Stalham,
  Norfolk.
  0692-80410.
- Stephen Hart,
  42 Midship Point,
  Malabar Street,
  London E14.
- David Phillips,
  49 Hartfield Avenue,
  Elstree,
  Herts.

- WO2 M.G. Collins,
  653 Sqn., 3 Regt. AAC,
  BFPO 106.
  02927-8-721
- B. Debenham,
  7 Beulah Grove,
  Leeds LS6.
  (Polytechnic term time only)
- Clement Ho,
  William Harvey Hospital,
  Ashford,
  Kent TN24 0LZ.
- John Wardley,
  Pleasure Park,
  Barry Island,
  South Glam. CF6 8ZD.
  0446 732844/738895
- R. Haynes,
  73 Huron Road,
  London SW17 8RG.
  01-672 8376
- L.S. Fisher.
  0227 65948
- Allan Honnor,
  17 Dene Gardens,
  Thames Ditton,
  Surrey KT7 0BJ.
- D. Clarke,
  Pipers,
  Green Road,
  Woolpit,
  Bury St. Edmunds,
  Suffolk IP30 9RF.

bookreview

The book Assembly Language Programming for the BBC Microcomputer by Ian Birnbaum seems to have taken off very well judging by the number of reviews we have seen on it. Last month Dr. Susans gave us his opinion on the book, this month we pass over to David Prideaux. Please do not send us any more reviews on this book as we will not be featuring it on this page again for some time to come.

Assembly language programming for the BBC Microcomputer by Ian Birnbaum £8.95 Macmillan Press.

The first impression of this book is that its 305 pages are packed with information: there is little in the way of blank pages or double spacing (or even cartoons). Far from counting against it, however, I found this an indication of value for money.

The author assumes the reader is familiar with BASIC but has no real knowledge of machine code or of the operation of a computer at a theoretical level. As this matches exactly my position, I found it precisely what I have been waiting for. Starting from scratch, he introduces the concepts involved in a logical manner, providing exercises at frequent intervals to test the reader's progress. The section giving the answers to the exercise occupies some 50 pages, so there is little chance of the serious student being disappointed.

Demonstration programs are freely distributed throughout the text, both illustrating the points under consideration and being useful into the bargain. The total of 73 listings encompasses some most desirable utilities, including a high resolution screen dump to Epson printers which copies a Mode 4 screen in 62 seconds. This compares with about 20 minutes for the BASIC equivalent and makes an excellent argument in its right for the advantages of machine code, showing the quoted increase in speed of a factor of about twenty. Several machine code sorting programs, a full machine code monitor and a program compactor, memory search utility and corrupt program retriever add up to good value and could justify buying the book just for them, ignoring the educational aspect.

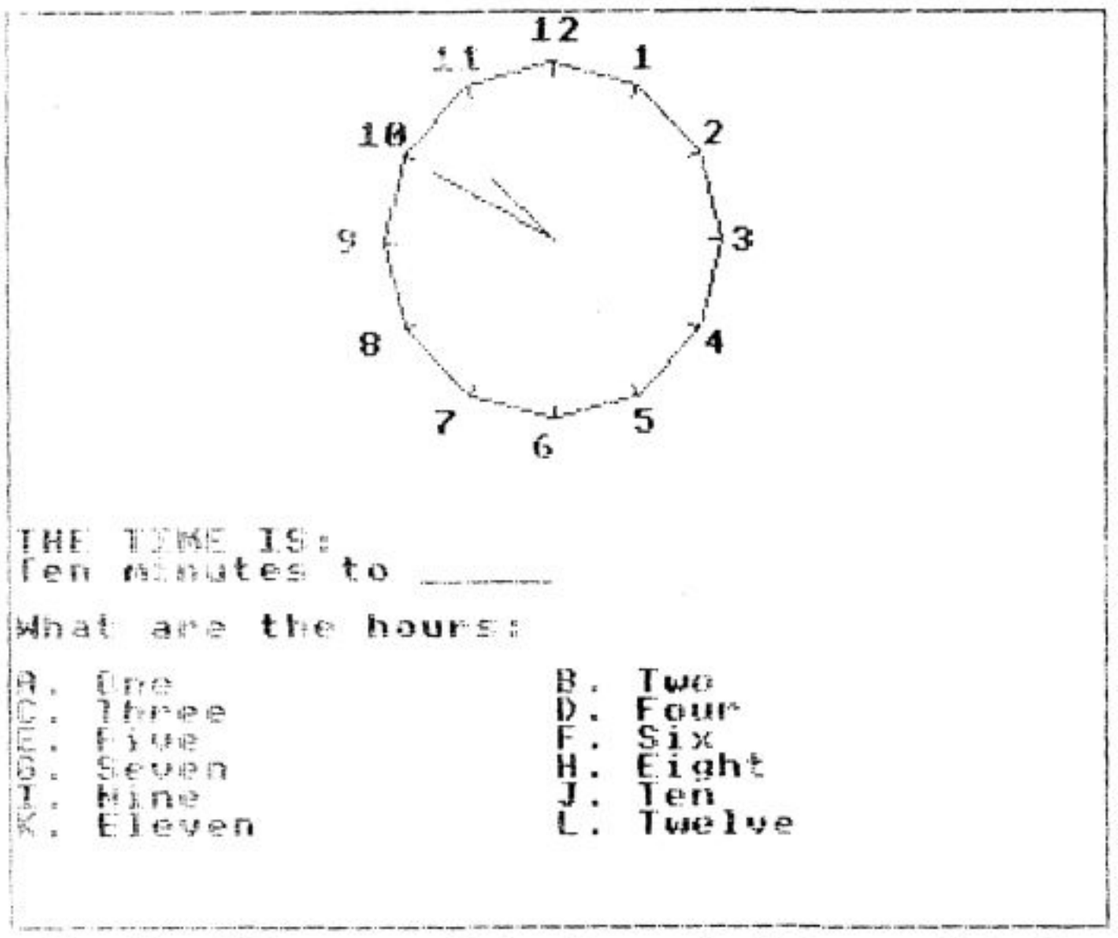
As well as covering the elements of the operation of machine code and the means of producing it with assembly language, which of course is appropriate to any computer which uses the 6502 chip, this book is BBC specific and consequently includes information on, for example, putting VDU statements into machine code.

Nine appendices include a summary of the full instruction set for the 6502 chip, a brief account of the 6522 chip (which operates the user port on Model B) and its operation under machine code instructions, some of the important zero page locations and notes on the smaller number of major differences between the 0.1 and later Operating Systems.

Associated with the book are two program cassettes, which between them hold all the listings from the book plus two further programs each, not given in the book: these are a graph plotter and a machine code disassembler (in BASIC) on one, a code location finder and "code find and replace" routine on the other. If you find keying-in programs frustrating then these cassettes are for you, but at £9 each (£16 the pair) I feel they are an expensive way of saving wear on my typing finger. I cannot understand how publishers can justify the price of program cassettes such as this, even with the four additional programs. Presumably people do buy them, however.

In summary, I found this book provides precisely what I need, as a relative newcomer to computing, to progress from BASIC to machine code and provides the key to unlocking yet another area of the BBC micro which I had hitherto barely glanced into. I look forward to spending some weeks getting to grips with all the information it contains. This book will certainly occupy an important place on my bookshelf adjacent to the computer.

David Prideaux



One thing children are taught in primary school is how to tell the time. However with the advent of digital watches it is surprising how many children cannot tell the time by anything other than a digital clock face.

The program below is hence of two-fold use. It gives the child ten different clock faces with which he must work out the correct time. To give an added sense of reality with the clock if the time is something to the hour then the hour hand moves to half way between the current hour and the coming one.

Your comments on our educational programs are always welcomed.

L.

```
 10 REM      What's The Time ?
 20 REM       by Paul Barbour
 30 :
 40 REM             14/1/83
 50 :
 60 REM          Version 1.0
 70 :
 80 REM Takes up ~5.07k memory
 90 REM (~8.92k with variables)
100 :
110 REM     Uses MODEs 4 & 7
120 :
130 REM  Designed for Primary
140 REM      School Children
150 :
160 REM       Requires 32k
170 :
180 REM     (c) LASERBUG 1983
190 :
200     :::::
210 :
220 MODE7
230 VDU23;8202;0;0;0;
240 ENVELOPE1,3,0,0,0,0,0,0,121,
-10,-5,-5,120,120
250 PROCtitle
260 PROCintro
270 MODE4
280 VDU23;8202;0;0;0;
290 VDU19,0,6,0,0,0,19,1,4,0,0,0
300 FOR0=1TO10
310   PROCclock
320   PROCtime
330   PROCquestion
340   PROCmark
```

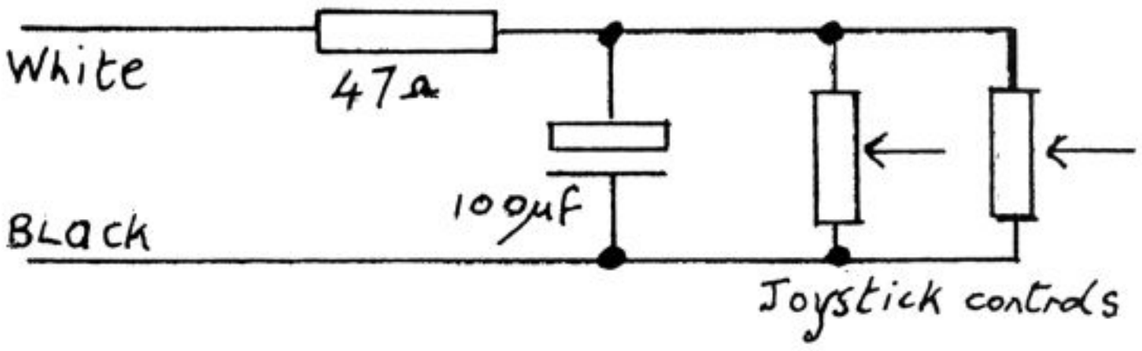
```
350  · CLS
360    NEXT
370 MODE7
380 VDU23;8202;0;0;0;
390 PROCscore
400 RUN
410 :
420     : : : : :
430 :
440 DEFPROCtitle
450 PRINTTAB(11,12);CHR$141;"WHA
T'S THE TIME ?"
   460 PRINTTAB(11,13);CHR$141;"WHA
T'S THE TIME ?"
   470 A%=INKEY(300)
   480 ENDPROC
   490 :
   500 DEFPROCintro
   510 CLS
   520 PRINT" "CHR$157;TAB(10);CHR$
141;CHR$129;"WHAT'S THE TIME ?"
   530 PRINT" "CHR$157;TAB(10);CHR$
141;CHR$129;"WHAT'S THE TIME ?"
   540 PRINTCHR$134;CHR$157;TAB(11)
;CHR$132;"(c) LASERBUG 1983"
   550 PRINT'TAB(9);CHR$130;"HICKOR
Y DICKORY DOCK,"
   560 PRINTTAB(6)CHR$130;"THE MOUS
E RAN UP THE CLOCK."
   570 PRINTTAB(9)CHR$130;"THE CLOC
K STRUCK ONE,"
   580 PRINTTAB(10)CHR$130;"THE MOU
SE RAN DOWN."
   590 PRINTTAB(9)CHR$130;"HICKORY
DICKORY DOCK."
   600 PRINT'CHR$131;"      This nur
sery rhyme was about the"'CHR$131;
"mouse and a clock but can YOU tel
l the"
   610 PRINTCHR$131;"time ?"
   620 PRINT'CHR$131;"      The comP
uter will test you by"'CHR$131;"sh
owing ten ""proPer"" clock faces.
 By"
   630 PRINTCHR$131;""proPer"" I m
ean that the clocks will"'CHR$131;
"have hands on,  not ""digital"" t
ime."
   640 PRINT'CHR$131;"      By answe
ring three questions can"'CHR$131;
"tell me what you think the time i
s."
   650 PRINT'CHR$131;"Press the";CH
R$129;"SPACE BAR";CHR$131;"to cont
inue..."
   660 REPEATUNTILINKEY(0)=32
   670 S%=0
   680 ENDPROC
   690 :
   700 DEFPROCclock
   710 VDU19,1,6,0,0,0
   720 MOVE640,968
   730 FORX%=0TO360STEP30
   740    DRAWSINRAD(X%)*200+640,COS
RAD(X%)*200+768
   750    NEXT
   760 FORX%=0TO330STEP30
   770    MOVESINRAD(X%)*200+640,COS
RAD(X%)*200+768
   780    DRAWSINRAD(X%)*185+640,COS
RAD(X%)*185+768
   790    NEXT
   800 VDU5
   810 C%=1
   820 FORX%=30TO360STEP30
   830    MOVESINRAD(X%)*235+615,COS
RAD(X%)*235+783
   840    PRINT;C%
   850    C%=C%+1
   860    NEXT
   870 VDU4
   880 ENDPROC
   890 :
   900 DEFPROCtime
   910 hours%=RND(12)
   920 screen_minutes%=RND(12)-1
   930 minutes%=screen_minutes%*5
   940 IFminutes%>=30THENadd%=15:EL
SEadd%=0
   950 screen_hours%=(hours%*30)+ad
d%
   960 MOVE640,768
   970 DRAWSINRAD(screen_hours%)*10
0+640,COSRAD(screen_hours%)*100+76
8
   980 MOVE640,768
   990 DRAWSINRAD(screen_minutes%*3
0)*160+640,COSRAD(screen_minutes%*
30)*160+768
  1000 VDU19,1,4,0;0;
  1010 ENDPROC
  1020 :
  1030 DEFPROCquestion
  1040 PRINTTAB(0,18);"Look careful
ly and decide what the time"'"is.
 When you think you know Press the
"'"RETURN key..."
  1050 REPEATUNTILINKEY(0)=13
  1060 PRINTTAB(0,18);"THE TIME IS:
";SPC(30)
  1070 PRINTTAB(0,19)"_____ _____ -
_____";SPC(120)
  1080 PRINTTAB(0,21)"Is it:"
  1090 PRINT'"A. something Past the
 hour";SPC(10)
  1100 PRINT"B. something To the ho
ur";SPC(10)
  1110 PRINT"C. something O'Clock";
SPC(10)
  1120 FORL=1TO3:PRINTSPC(30):NEXT
```

```
 1130 REPEATA%=GET:UNTILA%>=65ANDA
%<=67
 1140 IFA%=65THENrough$="+":answer
$="_____ Past _____":middle$="Past
"
 1150 IFA%=66THENrough$="-":answer
$="_____ to _____":middle$="to"
 1160 IFA%=67THENrough$="O":answer
$="_____ O'Clock":middle$="O'Clock
":guess_minutes%=0
 1170 PRINTTAB(0,19);answer$;SPC(3
0)
 1180 IFrough$="O"THENbegin$="":GO
TO1310
 1190 PRINTTAB(0,21);"How many min
utes is it ";middle$;" the hour:";
SPC(160)
 1200 PRINTTAB(0,23);"A. Five";TAB
(20);"B. Ten"
 1210 PRINT"C. Quarter";TAB(20);"D
. Twenty"
 1220 PRINT"E. Twenty-five";TAB(20
);"F. Half"
 1230 REPEATA%=GET:UNTILA%>=65ANDA
%<=70
 1240 IFA%=65THENguess_minutes%=5:
begin$="Five minutes"
 1250 IFA%=66THENguess_minutes%=10
:begin$="Ten minutes"
 1260 IFA%=67THENguess_minutes%=15
:begin$="Quarter"
 1270 IFA%=68THENguess_minutes%=20
:begin$="Twenty minutes"
 1280 IFA%=69THENguess_minutes%=25
:begin$="Twenty-five minutes"
 1290 IFA%=70THENguess_minutes%=30
:begin$="Half"
 1300 IFrough$="-"THENguess_minute
s=(30-guess_minutes)+30
 1310 IFbegin$=""THENPRINTTAB(0,1?
);"_____ O'Clock";SPC(140):GOTO133
0
 1320 PRINTTAB(0,19);begin$;" ";mi
ddle$;" _____";SPC(140)
 1330 PRINTTAB(0,21);"What are the
 hours:";SPC(255)
 1340 PRINTTAB(0,23);"A. One";TAB(
20);"B. Two"
 1350 PRINT"C. Three";TAB(20);"D.
Four"
 1360 PRINT"E. Five";TAB(20);"F. S
ix"
 1370 PRINT"G. Seven";TAB(20);"H.
Eight"
 1380 PRINT"I. Nine";TAB(20);"J. T
en"
 1390 PRINT"K. Eleven";TAB(20);"L.
 Twelve"
 1400 REPEATA%=GET:UNTILA%>=65AND9
%<=76
 1410 IFA%=65THENhours$="One"
 1420 IFA%=66THENhours$="Two"
 1430 IFA%=67THENhours$="Three"
 1440 IFA%=68THENhours$="Four"
 1450 IFA%=69THENhours$="Five"
 1460 IFA%=70THENhours$="Six"
 1470 IFA%=71THENhours$="Seven"
 1480 IFA%=72THENhours$="Eight"
 1490 IFA%=73THENhours$="Nine"
 1500 IFA%=74THENhours$="Ten"
 1510 IFA%=75THENhours$="Eleven"
 1520 IFA%=76THENhours$="Twelve"
 1530 guess_hours%=A%-64
 1540 IFrough$="O"THENPRINTTAB(0,1
9);hours$;" ";middle$;SPC(255);SPC
(150):GOTO1560
 1550 PRINTTAB(0,19);begin$;" ";mi
ddle$;" ";hours$;SPC(255);SPC(150)
 1560 IFrough$="-"THENguess_hours%
=guess_hours%-1:IFguess_hours%=-1T
HENguess_hours%=12
 1570 ENDPROC
 1580 :
 1590 DEFPROCmark
 1600 SOUND1,1,100,10
 1610 PRINTTAB(0,21)"You ";
 1620 PROCsec
 1630 SOUND1,1,125,10
 1640 PRINT"are ";
 1650 PROCsec
 1660 IFrough$="-"THENguess_minute
s%=(30-guess_minutes%)+30
 1670 IFminutes%=guess_minutes%AND
hours%=guess_hours%THENPROCright:E
LSEPROCwrong
 1680 TIME=0:REPEATUNTILTIME=200
 1690 ENDPROC
 1700 :
 1710 DEFPROCsec
 1720 TIME=0:REPEATUNTILTIME=150
 1730 ENDPROC
 1740 :
 1750 DEFPROCright
 1760 PRINT''TAB(15)"**********"
 1770 PRINTTAB(15)"*        *"
 1790 PRINTTAB(15)"* RIGHT *"
 1800 PRINTTAB(15)"**********"
 1810 FORX=1TO255
 1820   SOUND&11,1,X,1
 1830   NEXT
 1840 S%=S%+1
 1850 ENDPROC
 1860 :
 1870 DEFPROCwrong
 1880 PRINT''TAB(15)"**********"
 1890 PRINTTAB(15)"*        *"
 1900 PRINTTAB(15)"* WRONG *"
 1910 PRINTTAB(15)"*        *"
 1920 PRINTTAB(15)"**********"
 1930 FORX=255TO1STEP-1
```

continued on page 17

With the BBC Joysticks it has been found that, if the outputs are scales so as to give a full screen coverage, then there can be appreciable jitter on the position of the selected point even when the joysticks are not touched. This jitter can be removed by a suitable software filter, but this will slowdown operation, often to an undesirable extent.

The jitter is caused by poor smoothing of the power supply to the joysticks. It is possible to incorporate a simple RC filter in each of the joysticks which will reduce this jitter by about 8 times. Any residual jitter is then rarely of consequence. This modification does not change the speed of response in any way.



To carry out the modification, the case of the joystick is opened by undoing the four screws. A 47 ohm resistor is then inserted in the positive lead (coloured white) as near to the potentiometers as is practical. A 100 uf (greater than/equal to 3V working) capacitor is then connected across the potentiometers with the negative lead to the black wire and the positive lead to the 47 ohm resistor. All of the joints should be soldered so as to avoid any additional jitter being caused by bad connections.

This modification reduces the range of the outside by about 1%, a negligible amount and less than the jitter which would otherwise occur.

When using joysticks it can be difficult to accurately position a point anywhere on the screen, the following program shows one way of overcoming this problem by using a mixture of position and velocity control. In use, the controls can give linear operation over the centre half of their travels. Moving the joysticks further moves the centre of operation of the linear region and thus the displayed point. The further over the joystick is moved, the faster the movement.

In the program A% ( ) = ADVAL values
X%, Y% = screen coordinates
XY% = values of screen linear range centre position.
Lines 80,100 DIV11 gives velocity range. Reducing 11 speeds up movement
Lines 130, 150 DIV45 gives X scaling
Lines 140, 160 DIV58 gives Y scaling
Lines 90, 170 to 210 give limit stops
Line 210 checks if joystick button is pressed, if so then DRAW a line otherwise MOVE with current point marked on screen.
Line 230 detects DELETE key, and if pressed whilst the joystick is moved, it will invert the colour at this point thus deleting it. This key does require some practice to operate satisfactorily.
Line 250 clears the screen when the SPACE BAR is pressed.
Line 260 changes colour (randomly) if shift is pressed.

It is hoped that this program will give some ideas for the easy and successful use of joysticks.

Dr D E Susans

```
 1REM JOYSTICK D+V D.E.SUSANS (C) 1982
 10MODE2
 20 DIM A%(4),XY%(4),X%(2),Y%(2),OX%(2),OY%(2)
 30VDU23;11,0;0;0;0
 40OX%(1)=X%(1):OX%(2)=X%(2)
 50OY%(1)=Y%(1):OY%(2)=Y%(2)
 60FOR B%=1 TO 4
 70A%(B%)=ADVAL(B%)
 80 IF A%(B%)<&4000 XY%(B%)=XY%(B%)+(A%(B%)-&4000)DIV11
 90 IFXY%(B%)<0 THEN XY%(B%)=0
100 IF A%(B%)>&C000 XY%(B%)=XY%(B%)+(A%(B%)-&C000)DIV11
110 IFXY%(B%)>&FFFF THEN XY%(B%)=&FFFF
120NEXT
130X%(1)=(XY%(1)+(A%(1)-&8000)DIV6)DIV45
140Y%(1)=(XY%(2)+(A%(2)-&8000)DIV6)DIV58
150X%(2)=(XY%(3)+(A%(3)-&8000)DIV6)DIV45
160Y%(2)=(XY%(4)+(A%(4)-&8000)DIV6)DIV58
170FOR B%=1TO2
180IFX%(B%)<0 X%(B%)=0
190IFX%(B%)>1279 X%(B%)=1279
200IFY%(B%)<0 Y%(B%)=0
210IFY%(B%)>1023 Y%(B%)=1023
220IFADVAL(0)ANDB%GCOL1,B%: MOVE OX%(B%),(1023-OY%(B%))
:DRAW X%(B%),(1023-Y%(B%)) ELSEGCOL3,B%:PLOT 69,OX%(B%),(
1023-OY%(B%)):PLOT 69,X%(B%),(1023-Y%(B%))
230IF INKEY(-90) GCOL0,128:PLOT 69,X%(B%),(1023-Y%(B%))

240NEXT
250IF INKEY(-99) CLS
260IF INKEY(-1) VDU19,1,RND(7),0;0,19,2,RND(7),0;0
270GOTO40
```

debug. aid for prog. in mode 2

A notion I've found useful for debugging is to tack on at the end of your program a DEFPROCdbg(Q,QQ). It's then very simple to inject or delete PROCdbg(n,nn) anywhere in the program to inspect the current state of two (more if you wish) variables without ending the run. It is particularly useful under VDU5 when you have a screenful of graphics, etc. as you can create a window for the results (but remember that for printing numerals in the window the X in MOVEX,Y has to be 10 "text-cells" to the left of the left edge of the place you want the rightmost digit to appear! Maybe this is what confused your correspondent in LASERBUG 6 – apart of course from your deliberate (just-to-see-if-we're-all-awake) mistake!

The Proc I'm currently using (in MODE 2) is:

```
10000 DEFPROCdbg(Q,QQ)
10001 VDU24,80;14;480;78;16
10002 VDU25,4,-350;42;:PRINTQ",";QQ
:VDU26
10003 G=GET
10004 ENDPROC
```

Patrick Dowling

In this fact all action program for both the Models A and B you must race your car over a 20 mile course without crashing. Good luck all you budding James Hunts out there.

```
LIST
    10 REM    LASERBUG GRAND PRIX
    20 :
    30 REM       by Paul Barbour
    40 :
    50 REM          9/11/82
    60 :
    70 REM        Version 1.0
    80 :
    90 REM  Takes up ~3.96k memory
   100 :
   110 REM    and uses MODES 4 & 7
   120 :
   130 REM      Suitable for 16k
   140 :
   150 REM Written on OS EPROM 0.10
   160 :
   170 REM    (C) LASERBUG 1982
   180 :
   190     :::::
   200 :
   210 MODE7
   220 VDU23;8202;0;0;0;
   230 PROCinstructions
   240 MODE4
   250 VDU23;8202;0;0;0;
   260 VDU19,0,4,0,0,0
   270 PROCdefine_characters
   280 PROCdefine_variables
   290 PROCset_up_screen
   300 REPEATPROCmove_car
   310    PROCdata
   320    PROCcheck_for_crash
   330    PROCdraw_car
   340    PROCdata
   350    PROCmove_Path
   360    PROCdata
   370    UNTILmiles>=20
   380 PROCfanfare
   390 PROCwin
   400 PROCdata_and_another_game
   410 :
   420     :::::
   430 :
   440 DEFPROCdefine_characters
   450 VDU23,224,24,90,126,90,24,90
,126,90
   460 VDU23,225,255,195,165,153,15
3,165,195,255
   470 VDU23,226,-1,-1,-1,-1,-1,-1,
-1,-1
   480 ENDPROC
   490 :
   500 DEFPROCdefine_variables
   510 car_x=20
   520 path_x=15
   530 miles=0
   540 A%=135
   550 :
   560 DEFPROCset_up_screen
   570 FORX=1TO32
   580    PRINTTAB(15);CHR$225;TAB(2
5);CHR$225
   590    NEXT
   600 PRINTTAB(car_x,15);CHR$224
   610 TIME=0
   620 ENDPROC
   630 :
   640 DEFPROCmove_car
   650 PRINTTAB(car_x,15);" "
   660 IFINKEY(-26)THENcar_x=car_x-
.6:TIME=TIME+10
   670 IFINKEY(-122)THENcar_x=car_x
+.6:TIME=TIME+10
   680 :
   690 DEFPROCmove_Path
   700 path_x=Path_x+((RND(3)-2)/1)
   710 IFPath_x<0THENPath_x=0
   720 IFPath_x>28THENPath_x=28
   730 PRINTTAB(path_x,31);CHR$225;
TAB(Path_x+10,31);CHR$225
   740 PRINTTAB(Path_x,31);CHR$225;
TAB(Path_x+10,31);CHR$225
   750 miles=miles+0.04
   760 ENDPROC
   770 :
   780 DEFPROCcheck_for_crash
   790 PRINTTAB(car_x,17);
   800 IF(USR(&FFF4)AND&FF00)/&100=
225THENPROCbang:PROCcrash:PROCdata
_and_another_game
   810 ENDPROC
   820 :
   830 DEFPROCdraw_car
   840 PRINTTAB(car_x,17);CHR$224
   850 ENDPROC
   860 :
   870 DEFPROCdata
   880 distance=(INT((miles*10)+0.5
))/10
   890 time=TIME
   900 PRINTTAB(0,0);"DISTANCE: ";d
istance;" miles | TIME: ";INT(time
/100);" secs"
   910 ENDPROC
   920 :
   930 DEFPROCinstructions
   940 PRINTCHR$157
   950 PRINTCHR$157TAB(8)CHR$141;CH
R$129"LASERBUG GRAND PRIX"
   960 PRINTCHR$157;TAB(8)CHR$141;C
HR$132"LASERBUG GRAND PRIX"
   970 PRINTCHR$157
   980 PRINT'CHR$134;" In this game
```

```
 you take on the role of"
  990 PRINT'CHR$134;" a Formula 1
driver competing in a 20"
 1000 PRINT'CHR$134;" lap race wit
h each lap being 1 mile"
 1010 PRINT'CHR$134;"long.  You mu
st complete the course in"
 1020 PRINT'CHR$134;"  as little t
ime as possible and of"
 1030 PRINT'CHR$134;"  course wit
hout crashing into the"
 1040 PRINT'CHR$134;"  barrier wh
ich marks your course."
 1050 PRINT'CHR$133"Use";CHR$130;"
C";CHR$133;"to move left and";CHR$
130;"J";CHR$133;"to move right."
 1060 PRINTCHR$131"      (Top right
 of the keyboard)"
 1070 PRINT'CHR$157;CHR$132"   Pre
ss";CHR$129;"RETURN";CHR$132;"to s
tart game...."
 1080 REPEATUNTILINKEY(-74)
 1090 ENDPROC
 1100 :
 1110 DEFPROCbang
 1120 VDU19,0,7,0,0,0
 1130 PRINTTAB(car_x,17);CHR$225
 1140 VDU19,1,1,0,0,0
 1150 VDU19,0,4,0,0,0
 1160 VDU19,0,7,0,0,0
 1170 PRINTTAB(car_x,17);CHR$224
 1180 VDU19,1,7,0,0,0
 1190 VDU19,0,4,0,0,0
 1200 FORdelay=1TO500
 1210   NEXT
 1220 FORexplosion=-15TO0STEP.1
 1230   VDU19,1,1,0,0,0
 1240   SOUND0,explosion,6,1
 1250   VDU23,224,RND(255),RND(255
),RND(255),RND(255),RND(255),RND(2
55),RND(255),RND(255)
 1260   PRINTTAB(car_x,17);CHR$224
 1270   VDU19,1,7,0,0,0
 1280   NEXT
 1290 ENDPROC
 1300 :
 1310 DEFPROCcrash
 1320 VDU22,7
 1330 VDU23;8202;0;0;0;
 1340 PRINTCHR$157
 1350 PRINTCHR$157TAB(8)CHR$141;CH
R$129"LASERBUG GRAND PRIX"
 1360 PRINTCHR$157;TAB(8)CHR$141;C
HR$132"LASERBUG GRAND PRIX"
 1370 PRINTCHR$157
 1380 PRINT''CHR$134"You crashed i
nto the barrier..."
 1390 FORdelay=1TO4000
 1400   NEXT
 1410 PRINTCHR$133"...your car exp
loded..."
 1420 FORdelay=1TO4000
 1430   NEXT
 1440 PRINTCHR$130"...and you were
 pulled out of the"
 1450 PRINTCHR$130"reckage..."
 1460 FORdelay=1TO4000
 1470   NEXT
 1480 PRINTCHR$131"...still alive
and kicking thankfully !"
 1490 ENDPROC
 1500 :
 1510 DEFPROCdata_and_another_game
 1520 PRINTCHR$134"You traveled ";
(INT((miles*10)+0.5))/10;" miles"
 1530 PRINTCHR$134"in a time of ";
time/100;" seconds"
 1540 PRINT''CHR$157;CHR$129;" Ano
ther Game ? ";CHR$132;"(Press ""Y""
 or ""N"")"
 1550 REPEATUNTIL(INKEY(-69)ORINKE
Y(-86))
 1560 IFINKEY(-69)THENRUN
 1570 *FX15
 1580 END
 1590 :
 1600 DEFPROCfanfare
 1610 SOUND1,-15,100,10
 1620 SOUND2,-15,101,10
 1630 SOUND3,-15,102,10
```

continued on page 18

continued from page 13

```
 1940    SOUND&11,1,X,1
 1950    NEXT
 1960 ENDPROC
 1970 :
 1980 DEFPROCscore
 1990 PRINTCHR$131;"You have finis
hed the test and scored:"
 2000 PRINT'TAB(12)CHR$141;CHR$130
;S%;" out of 10"
 2010 PRINTTAB(12)CHR$141;CHR$130;
S%;" out of 10"
 2020 IFS%=10THENC$="That is excel
lent."
 2030 IFS%>=8ORS%=9THENC$="That is
very good."
 2040 IFS%=6ORS%=7THENC$="That is
quite good."
 2050 IFS%<=5THENC$="You could hav
e done better !"
 2060 PRINT'CHR$131;C$
 2070 PRINT''CHR$134;"Next person
press";CHR$129;"RETURN";CHR$134;"t
o start or"
 2080 PRINTCHR$129;"E";CHR$134;"to
end."
 2090 A=GET
 2100 IFA=13THENENDPROC
 2110 IFA<>69THEN2090
```

nine dice/roll dem bones

Below is a program from John Claydon which performs the game (???) "Roll 'dem bones". The program works entirely in teletext mode but does anybody have any idea of the rules (or more to the point are there any?) Well, whatever the idea of it all it looks good at least!

L.

```
  10 REM      "ROLL DEM BONES"
  20 REM       by John Claydon
  30 REM of the BBC MicrocomPuter
  40 REM    Education WorkshoP.
  50 :
  60 REM    Meets on the second
  70 REM  Sunday afternoon each
  80 REM  month at Bounds Green
  90 REM    Junior School,  Park
 100 REM    Road,  London N.11.
 110 REM Phone 889-5446 for more
 120 REM  details or write to 10
 130 REM Barnes Court,  Clarence
 140 REM Road,  London,  N22 4PJ.
 150 :
 160 REM         Version 1.0
 170 :
 180 REM  Takes uP ~2.00k memory
 190 REM   and uses MODE 7 only
 200 :
 210 REM      Suitable for 16k
```

continued on page 18

continued from page 7

```
  840
  850
  860:::::::::::::::::::::::::::::::::
:::::
  870
  880DEFPROCPLAY
  890
  900   @%=&202: VDU4
  910   PRINT TAB(4,2) Ball%
  920   @%=&404: VDU5
  930
  940 REPEAT: PROCBAT: UNTIL ADVAL
0AND3
  950
  960   X%=0:  Y%=512
  970   U%=16: R%=0
  980   V%=(M%+1-RND(M%DIV2))*SGN(R
ND)
  990   GCOL3,7: MOVEX%,Y%: VDU226
 1000
 1010REPEAT: PROCBAT
 1020
 1030   FOR D%=1 TO M%DIV4+1
 1040   O%=X%: P%=Y%
 1050   X%=X%+U%: Y%=Y%+V%
 1060   Z%=Y%+V%: IF Z%>903 OR Z%<1
12 V%=-V%: SOUND&12,1,50,5
 1070   IF X%>K%-33 AND X%<736 THEN
 1150
 1080   Z%=Y%-L%: IF Z%>120 OR Z%<0
 THEN 1110
 1090   IF X%>I%-64 AND X%<K%-32 U%
=ABS(U%): X%=K%: C%=7: SOUND0,-9,4
,1: PROCZAP
 1100
 1110   F%=X%+52+16*SGN(U%)
 1120   C%=POINT(F%,Y%-10)
 1130   IF C% AND C%<8 U%=-U%: PROC
ZAP
 1140
 1150   GCOL3,7: MOVE O%,P%: VDU226
: MOVE X%,Y%: VDU226
 1160   NEXT
 1170
 1180UNTIL X%<-32 OR S%=A%
 1190
 1200   GCOL3,7: MOVE X%,Y%: VDU226
 1210   IF S%=A% A%=A%+1000: Ball%=
Ball%-1: PROCOURT: ENDPROC
 1220   REM S%=S%-R%: PROCSCOR
 1230   REM PROCWALL(1): R%=0
 1240
```

continued from page 16

```
1640 SOUND1,-15,104,10
1650 SOUND2,-15,105,10
1660 SOUND3,-15,106,10
1670 SOUND1,-15,108,10
1680 SOUND2,-15,109,10
1690 SOUND3,-15,110,10
1700 SOUND1,-15,112,10
1710 SOUND2,-15,113,10
1720 SOUND3,-15,114,10
1730 SOUND1,-15,116,10
1740 SOUND2,-15,117,10
1750 SOUND3,-15,118,10
1760 SOUND1,-15,108,10
1770 SOUND2,-15,109,10
1780 SOUND3,-15,110,10
1790 SOUND1,-15,100,20
1800 SOUND2,-15,101,20
1810 SOUND3,-15,102,20
1820 FORdelay=1TO8000
1830   NEXT
1840 ENDPROC
1850 :
1860 DEFPROCwin
1870 VDU22,7
1880 VDU23;8202;0;0;0;
1890 PRINTCHR$157
1900 PRINTCHR$157TAB(8)CHR$141;CH
R$129"LASERBUG GRAND PRIX"
1910 PRINTCHR$157;TAB(8)CHR$141;C
HR$132"LASERBUG GRAND PRIX"
1920 PRINTCHR$157
1930 PRINT''CHR$134"You scream ov
er the finish line..."
1940 FORdelay=1TO4000
1950   NEXT
1960 PRINTCHR$133"...and deafened
by the crowds cheering"
1970 PRINTCHR$133"you..."
1980 FORdelay=1TO4000
1990   NEXT
2000 PRINTCHR$131"...you collect
the Championship cup..."
2010 FORdelay=1TO4000
2020   NEXT
2030 PRINTCHR$130"...and are over
welmed with"
2040 PRINTCHR$130"congratulations
!"'
2050 ENDPROC
```

Paul Barbour

continued from page 17

```
  220 :
  230 REM Written on OS EPROM 0.10
  240 :
  250 REM     (c) LASERBUG 1982
  260 :
  270      :::::
  280 :
  290 MODE 7
  300 PRINT''''TAB(8);:VDU 132,15
7,131,141;:PRINT "ROLL DEM BONES
 ";:VDU 156
  310 PRINTTAB(8);:VDU 132,157,131
,141;:PRINT "ROLL DEM BONES   ";:V
DU 156
  320 PRINT''''TAB(6);:VDU 129,15
7,135;:PRINT"PRESS RETURN TO ROLL
```

continued on page 19

continued from page 18

```
";:VDU 156
  330 REPEATUNTILGET=13:CLS
  340  DIM C(3)
  350  FOR A=1 TO 3
  360    FOR D=1 TO 3:LET C(D)=RND
(6):NEXTD
  370    FOR B=1 TO 3
  380      E=RND(7)
  390      PRINT TAB(A*13-12,B*8-7
);:VDU (144+E),32,120,124,124,124,
124,124,124,124,116,32,32
  400      PRINT TAB(A*13-12,B*8-5
);:VDU (144+E),106,255,255,255,255
,255,255,255,255,255,53
  410      PRINT TAB(A*13-12,B*8-3
);:VDU (144+E),106,255,255,255,255
,255,255,255,255,255,53
  420      ON C(B) GOTO 430,470,49
0,510,530,550
  430      PRINT TAB(A*13-12,B*8-6
);:VDU (144+E),106,255,255,255,255
,255,255,255,255,255,53
  440      GOSUB 570
  450      PRINT TAB(A*13-12,B*8-2
);:VDU (144+E),106,255,255,255,255
,255,255,255,255,255,53
  460      GOTO 660
  470      GOSUB 620
  480      GOTO 590
  490      GOSUB 570
  500      GOTO 590
  510      GOSUB 620
  520      GOTO 640
  530      GOSUB 570
  540      GOTO 640
  550      PRINT TAB(A*13-12,B*8-4
);:VDU (144+E),106,255,96,255,255,
255,255,255,96,255,53
  560      GOTO 640
  570      PRINT TAB(A*13-12,B*8-4
);:VDU (144+E),106,255,255,255,255
,96,255,255,255,255,53
  580      RETURN
  590      PRINT TAB(A*13-12,B*8-6
);:VDU (144+E),106,255,96,255,255,
255,255,255,255,255,53
  600      PRINT TAB(A*13-12,B*8-2
);:VDU (144+E),106,255,255,255,255
,255,255,255,96,255,53
  610      GOTO 660
  620      PRINT TAB(A*13-12,B*8-4
);:VDU (144+E),106,255,255,255,255
,255,255,255,255,255,53
  630      RETURN
  640      PRINT TAB(A*13-12,B*8-6
);:VDU (144+E),106,255,96,255,255,
255,255,255,96,255,53
  650      PRINT TAB(A*13-12,B*8-2
);:VDU (144+E),106,255,96,255,255,
255,255,255,96,255,53
  660      PRINT TAB(A*13-12,B*8-1
);:VDU (144+E),34,111,255,255,255,
255,255,255,255,63,33
  670      NEXT B
  680    NEXT A
  690 REPEATUNTILGET=13
  700  CLS:GOTO 350
```

```
1250ENDPROC
1260
1270
1280:::::::::::::::::::::::::::::::
::::::
1290
1300DEFPROCMORE
1310
1320  VDU4,23;8202;0;0;0;
1330  PRINT TAB(1,30) "Another ga
me? Y/N"
1340  IF INKEY-69 RUN
1350  PROCBAT
1360  IF NOT INKEY-86 THEN 1340
1370  FOR Z%=1 TO 500: NEXT
1380  CLEAR: MODE7: *FX15,0
1390
1400END
1410
1420
1430:::::::::::::::::::::::::::::::
::::::
1440
1450DEFPROCBAT
1460
1470  I%=K%: J%=L%
1480  K%=496-ADVAL(1)DIV2048*16
1490  L%=96+ADVAL(2)DIV256*8
1500  IF L%>803 L%=803
1510  IF I%=K% AND J%=L% ENDPROC
1520  GCOL3,7
1530  MOVEI%,J%: DRAWI%,J%+96
1540  MOVEK%,L%: DRAWK%,L%+96
1550
1560ENDPROC
1570
1580
1590:::::::::::::::::::::::::::::::
::::::
1600
1610DEFPROCZAP
1620
1630  IF C%=1 R%=R%+2
1640  IF C%=6 U%=-U%: ENDPROC
1650  IF C%=7 THEN 1780
1660  IF C%>0 THEN 1710
1670  IF X%<32 U%=-U%
1680  IF X%>1200 SOUND&12,1,50,5
1690ENDPROC
1700
1710  SOUND1,-13,200-C%*12,2
```

continued over

```
 1720   GCOL0,0
 1730   MOVE F%-32,32*((Y%+18)DIV32
)
 1740   VDU224
 1750   S%=S%+C%+1: PROCSCOR
 1760ENDPROC
 1770
 1780   Z%=96+ADVAL(2)DIV256*8
 1790   IF Z%>803 Z%=803
 1800   V%=V%+(Z%-J%)DIV10
 1810   IF V%=0 V%=V%+RND(3)-2
 1820   IF ABS(V%)>M% V%=M%*SGN(V%)
 1830   SOUND0,-9,4,1
 1840ENDPROC
 1850
 1860
 1870:::::::::::::::::::::::::::::::::
:::::
 1880
 1890DEFPROCWALL(C%)
 1900
 1910   GCOL0,C%
 1920   FOR Y%=4 TO 28
 1930    MOVE Wall%,Y%*32
 1940    VDU225
 1950   NEXT
 1960
 1970ENDPROC
 1980
 1990
 2000:::::::::::::::::::::::::::::::::
:::::
 2010
 2020DEFPROCSCOR
 2030
 2040   VDU4
 2050   PRINTTAB(15,2) S%
 2060   VDU5
 2070
 2080ENDPROC
```

Peter Voke

memory dump

The program below is fairly self explanatory. It provides a complete memory dump and gives memory location, what is held there (both in hex), the ASCII character in that location (if it is >32 or <127) and the value of that location in binary (!!!). You also have the option to list the results onto a printer (designed for parallel printers – add in the appropriate *FX5 if needed).

```
L.
   10 REM        Memory Dump
   20 REM     by Paul Barbour
   30 :
   40 REM          4/2/83
   50 :
   60 REM        Version 2.0
   70 :
   80 REM Takes up ~1.20k memory
   90 REM (4.87k with variables)
  100 :
  110 REM     Uses MODE 7 only
  120 :
  130 REM     Suitable for 16k
  140 :
  150 REM    Written on OS 0.1
  160 :
  170 REM    (c) LASERBUG 1983
  180 :
  190      :::::
  200 :
  210 ONERRORPRINT:VDU3:END
  220 MODE7
  230 VDU23;8202;0;0;0;
  240 PROCdata
  250 PROCheadings
  260 PROCmemory_dump
  270 END
  280 :
  290      :::::
  300 :
  310 DEFPROCdata
  320 PRINTCHR$134;"Enter start ad
dress:";CHR$131;"&";:INPUT""start$
  330 PRINTCHR$134;"Enter end addr
ess:";CHR$131;"&";:INPUT""finish$
  340 start%=EVAL("&"+start$)
  350 finish%=EVAL("&"+finish$)
  360 IFstart%<0ORstart%>&8000THEN
CLS:GOTO320
  370 IFfinish%<0ORfinish%>&8000TH
ENCLS:GOTO320
  380 IFfinish%<start%THENCLS:GOTO
320
  390 PRINTCHR$134;"Printer dump ?
 (Y/N)"
  400 REPEATprinter$=GET$:UNTILpri
nter$="Y"ORprinter$="N"
  410 ENDPROC
  420 :
  430 DEFPROCheadings
```

letters

Dear LASERBUG,

I really must write to comment on the last couple of issues of LASERBUG and in particular your two articles on software protection.

In the first article you mention how to find the start, end and execution addresses. However this is a very tiresome method and it is much easier to let the computer do it for you. Simply type *OPT2,0 before *LOADing the program and the computer will display the number of blocks, the length, the start address and the execution address of the program (all in hex). The copy can now be made in the manner described. *(The method of *OPT2,,0 might work but is incorrect. *OPT2,0 lets the computer ignore all errors though messages may be given" – ref p.398 of User Guide. The correct command is *OPT1,2 – see letters last month – Ed.)*

However my main point is that none of the methods so far described can prevent unauthorised copying by loading the program (or each program part in turn) and then saving them, without ever running the program(s). This method avoids all the software protection methods I have yet encountered and because of the *OPT2,0 *(or *OPT1,2 – Ed.)* function even programs with strange execution or loading addresses can be copied. *(Surely the method described in Software Protection Part I does not require you to run the program – Ed.)*

Admittedly actually listing a program can be made hard, but then since a great deal of good programs these days involve some machine code this will always be a problem.

continued on page 22

CONT can be implemented on the BBC by using GOTO ERL which will work unless the error occurred inside nested loops.

I was a bit disappointed that you had no adventure games reviewed in Issue 7, still will look forward to the promised review in Issue 8. However I would like to point out that there are now 35 (!!!) different adventures available for the BBC at the moment so I hope the review will cover most of them. *(Sorry but we didn't as you would have seen but we will in future review at least one adventure every month in softreview – Ed.)*

The ?216=32 does work for the 0.1 OS. *(We know that but does it work on series 1 OS – Ed.)*

Keep up the good work.

Nicholas Clifton,
Chislehurst, Kent, BR7 5HS.

Dear LASERBUG,

I subscribe to LASERBUG and was looking forward particularly to the Education Spot. Your "Maths Race" was nicely presented but of limited use in school. To cope with all levels of ability a) different difficulty levels and b) different content levels (e.g. addition only) are needed. If you could add these refinements to your program it could be used much more widely in schools. *(Generally we write an educational program for a set level – in this case it was aimed for the older children in a junior school such as yours to test their general ability. All our educational programs are written in a simple manner to allow them to be adapted for other uses. If you do modify Maths Race as you suggested then please let us have a copy and we will print it again in its new form – Ed.)*

In "Alphabet Tester" (1.2000) look out for "should **of**" and "practice" as a verb. *(Sorry teacher, I'll try to do better next time. Seriously though proof reading 24 pages AND the programs is a time consuming job and several mistakes always slip through. Believe it or not I passed my English exams at grade A standard – Ed.)*

Please don't misunderstand my motives. I value LASERBUG and as I'm a recent convert to micros I've learnt a lot from it. It's just that in schools we have to be so careful in making sure what we give children is accurate.

Mr. R.A. Smith, Headmaster,
Blue Bell Hill Junior School, Nottingham.

Dear LASERBUG,

This is to advise you of the formation of Wales' first BBC Microcomputer Club known as CBCC (Cardiff BBC Computer Club).

CBCC is fortunate in being able to hold its meetings (alternate Wednesday evenings) in the Applied Science Lecture Theatre of University College, Newport Road, Cardiff. Among other advantages the Lecture Theatre has four 24" elevated monitors, full audio visual equipment, dimmer and P.A. systems. These excellent facilities make this one of the finest venues available to any micro club.

CBCC started four months ago and already has over sixty members. It has connections with the computer industry and as a result has been able to demonstrate to its members items of both hardware and software ahead of general availability. A "Beginner's Corner" operates under the capable direction of Dr. David Wharry of South Glamorgan's Computer Education Department.

The main objectives of the club is to provide a forum for "Owners and Others" interested in the BBC Micro. Officials are Geoffrey Barker (Chairman), William Smith (Treasurer), Diana Edwards (Secretary) joined by Harry White and Peter James.

Geoff Barker, CBCC.

Dear LASERBUG,

Thank-you for your review of "Academy" in the December '82 Issue 7 of LASERBUG.

In reference to the review most companies produce a "Space Invader" type game and we followed suit. The listing of "Attack" from the first issue of the "Owl" was at hand when beginning to write our invader game Academy, because we felt Mark Cook's method of TABbing the aliens was succinct.

The routine itself being of only 10 lines and our final program with storyline, use of arrow keys and detailed scoring now lasting just over 200 lines, a breach of copyright was not even considered.

However to clarify this issue, I have sent a copy of our program to "Computer & Video Games" asking them if we should withdraw it until we have re-written the 10 line routine in question.

In connection with our other programs, all are completely original and this problem should not arise again.

I will let you know of "Computer & Video Games" answer, and will, if asked, withdraw the program completely from our catalogue, because the company does not want to breach copyright not cause any ill feeling between our software writers and writers of copyrighted material.

N.D. Munns, Swift Link Software,
London.
(Draw your own conclusions from the above letter – Ed.)

Dear LASERBUG,

The following is comment on articles in the LASERBUG Newsletter which you may like to edit and print, or incorporate in an article based on the opinions of other members of the club.

Having read the two articles in recent issues on Software Protection may I submit my opinion and concern regarding the consequences of such protection on Professional Software.

Very few programs purchased meet exact requirements, particularly data programs, wordprocessors and other more serious programs. It may be required to add extra features to suit a particular need, or, to save memory, delete some. If this is not possible, the program could be useless to the purchaser.

Even in games programs, it may be necessary to alter the listing to slow down or speed up the action. If this is not possible interest in the game soon wanes.

Any form of protection which precludes alteration is, in my opinion, not in the interest of the consumer.

May I hasten to add, I am against any form of piracy, and duplication for profit. The producers of programs have a need for some form of protection against being ripped off, though I have little sympathy for some of them whose programs, and their price, are a 'rip-off' anyway.

Thank-you for reading all this, some of which may give food for thought, and I look forward to the next LASERBUG issue and interesting reading.

Mr. W.S. Turner,
Hythe, Southampton.

Dear LASERBUG,

Regarding your article on changing operating systems (I think Issue 5) **ALL** software sold by IJK Software for the BBC Micro from 1st November '82 will run on **all** operating systems 0.1-1.2. Previous to this date, only a handful would not function correctly on the newer OS's and our policy here is to exchange these cassettes for anyone with problems for a nominal charge of £1.50 per cassette to cover handling and postage & packing.

Please note that users with Disc Interfaces automatically lose 2.75k of RAM – BASIC starts at &1900 with DFS. This may need to be recovered before CHAINing some of the longer programs (ours and others). This can be done by simply reseting page as follows:

```
*TAPE
PAGE=&E00
NEW
CHAIN" "
```

All the best for the New Year.

Ian Sinclair, IJK Software,
Blackpool, Lancs.

(The final letter was in fact drawn up after a telephone conversation. Unfortunately the person concerned had hurt his arm and was unable to write – Ed.)

Dear LASERBUG,

I have discovered a bug in the Acornsoft program METEORS. Sometimes when you shoot a big asteroid the missile goes straight through the asteroid and othertimes it goes through the edge facing you but does not destroy the asteroid until it hits the other side. Obviously this can mean the difference between life and death.

I returned the cassette to Acornsoft asking for a bug free copy but the copy they sent back was exactly the same. I have tried several other peoples versions of this tape and all have the same bug in it.

When I phoned up Acornsoft about this they got very annoyed about it and in the end they refused to speak to me. They said that they knew of this bug but if they cured it the missile would travel much slower. I have seen other companies versions of Asteroids but their missiles travel just as fast but do hit the asteroid every time. Hence this argument is nonsense and Acornsoft have no reason to leave the program uncorrected.

W.M. Dunning,
Potters Bar, Herts.
(Any comments Acornsoft ??? – Ed.)

---

continued from page 9

```
 1380    IFdata%(0,question%)=0ORda
ta%(i,question%)=0THENrating%=rati
ng%+1
 1390    NEXT
 1400 ENDPROC
 1410 :
 1420 DEFPROCresults
 1430 CLS
 1440 PRINTdble$;"SUITABILITY RATI
```

continued on page 23

continued from page 20

```
 440 CLS
 450 PRINTTAB(0,0);CHR$130;"MEMOR
Y   LOCATION   ASCII   BINARY"
 460 IFPrinter$="Y"THENVDU2
 470 PRINTTAB(0,23);CHR$130;"MEMO
RY   LOCATION   ASCII   BINARY"
 480 VDU28,0,22,39,1
 490 ENDPROC
 500 :
 510 DEFPROCmemory_dump
 520 FORmemory%=start%TOfinish%
 530    PRINTCHR$131;~memory%;TAB(
10);?memory%;TAB(21);
 540    IF?memory%>=32AND?memory%<
=126THENPRINTCHR$(?memory%);TAB(29
);:ELSEPRINTCHR$32;TAB(29);
 550    PROCbinary(?memory%)
 560    NEXT
 570 ENDPROC
 580 :
 590 DEFPROCbinary(decimal%)
 600 FORPower%=7TO0STEP-1
 610    binary%=2^Power%
 620    PRINT;decimal%DIVbinary%;
 630    decimal%=decimal%MODbinary
%
 640    NEXT
 650 PRINT
 660 ENDPROC
```

Paul Barbour

I read Mr Hirsts letter on p.17 with great interest on both points. I agree that the trimmer mentioned does effect the moving bars seen on a TV screen when the input is via the UHF socket. However in my experience and others (RCS and several Acorn agents) this adjustment only solves the problem a little if the trimmer was badly adjusted initially. If it was not then improvement only lasts a few minutes as it 'drifts off easily". This uncontrollable drifting which causes moving diagonal llines on a screen in varying degrees of size, speed and severity seems **directly** connected to the varying running temperature of the micro. Acorn advise nothing can be done. VERY HELPFUL (*aren't they always – Ed.*)

With regards to Mr Hirsts information concerning audio output on the micro I have acquired a different solution.

As I know very little about electronics I have taken peoples word for it that the connections are safe to use (one being Acorns technical dept.) All I can say is it works, has done on my micro for over three weeks and the only tiny side effect is an occasional quiet "electrical crackle" heard via the speakers of my hi-fi. This is either due to an unsuppressed electrical appliance or a tiny electrical surge or adjustment in the micro itself (*probably the first one – Ed.*)

I am informed that the point PL16 (Ref: User Guide, p. 503) which has two solder spots is suitable for connection via a lead to an AMP AUX input socket. Acorn have confirmed this personally to me and also advised that the voltage given off is 500mV.

A female phone socket can be neatly fitted in the side of the micro as indicated above and there is a convenient empty hollow in the micro case where no other circuitry is situated, this makes fitting easy and allows the shortest of wires from PL16 which minimizes any pickup of unshielded signals.

I have found the sound quality a fantastic improvement as you would expect through a hi-fi system and I assume a TV audio input would also be suitable.

My method of connection leaves the micro's own speaker fully connected and operative. A convenience when the micro is temporarily removed from its normal position.

David Glew

make the most out of sound

MAKE THE MOST OF SOUND by Patrick Dowling

Seperate queues for all 4 Channels, each holds up to 4 waiting requests. If queue if full, program waits. Queues are flushed independently. Unless H, S or F apply, next note sounds immediately previous one is completed (i.e. after D/20 seconds) 16 envelopes are available (series 1 OS only) but only 4 if BPUT# is used (4 maximum with 0.1 OS) Stored from &8C0 (series 1 OS)/&800 (0.1 OS)

```
                     (note: if pitch > 255,  Pitch MOD 256 is used
                     until it drops below 256 again - 'wraparound'.)      (note: only 16 Amplitude levels)
                                      P I T C H                                   A M P L I T U D E
          Env.   Step                                                    change of level
          number dur. size of steps(in units       no. of steps          per pitch step     target levels
          1 - 16 0-127 of freq.) -128 to +127        0 to 255            ± 127    -127 to 0   0 to 126
ENVELOPE   N   | T | PI1  |  PI2  | PI3  |  PN1  | PN2 | PN3 |        AA | AD | AS  |  AR  | ALA | ALD
        (see        (PI1+1)x(PN1+1)=Attack pitch rise starting from P      | Decay  |     |     |   Sustain start,
        note) x .01 (PI2+1)x(PN2+1)=Decay pitch change (from end of attack) | rate.  |     |     |   determines loudness
             (If +128(PI3+1)x(PN3+1)=Sustain pitch change (from end of decay) |(if zero, |  |     |   of the note
             prevents                                                  | level  |     |     |
             auto-       (T+1)x(PN1+1)=duration of Attack              | remains |    |    Attack peak-level
             repeat)     (T+1)x(PN2+1)=duration of Decay               | at ALA.)|    |
                         (T+1)x(PN3+1)=Duration of Sustain             |         |   Release rate -127 to 0 (if 0,  envelope
                            total/100 = duration of pitch-      Attack rate      |   auto repeats forever
H continuation control                  envelope; while <D/20 it   of change.    |
=1 allows previous note to continue     auto-repeats (T>127 inhibits          Sustain rate of change
dying away   |     Sync control         auto-repeat.)
(ignoring    |   =1,2,3 indicates how many
its own APD  |    |other notes are to be sounded simultaneously. - for making chords
parameters)  |    |     Flush control
- continuati|on   |     =1 sounds immediately & flushes queue,  overiding any current or waiting sounds - overide or dummy 'stopper note'.
'dummy note'|     |     |
             |    |     |    =0      produces 'noise' of type determined by Pitch (If P=0,1,2 - High/Med(=237)/Low(=189)frequency buzzing
             |    |     |    =1,2,3  channel number                            P=3     - buzz at freq. of Chan.1 pitch-setting.
             |    |     |      |                                               P=4,5,6 - High/Med(=237)/Low(=189)white noise
SOUND  &   H      S     F      C ,  A ,  P ,  D                                P=7     - ditto at freq. of Chan. 1 pitch setting.)
                                    |    |    |
                                    |    |   Duration (1/20) to end of Sustain phase (Note: determines moment next SOUND will start.)
                                    |    |
                                    |   Pitch (frequency) of sound; (or start of sound if envelope applied)
                                    |   0-255 (or 0-7 of Channel=0 see above)
                                    |
                                   Amplitude (volume) 0 to -15 of a plain,  unvaried sound
                                   or Envelope number 1 to 4 (up to 16 possible.)
```

credits

LASERBUG is edited by Paul Barbour.

The contributors this month were Paul Barbour, Patrick Dowling, Dr D E Susans, David Prideaux, Nick Goodwin and Peter Voke.

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 21

```
NG OF ";name$(0);" TO"
  1450 PRINTdble$;"SUITABILITY RATI
NG OF ";name$(0);" TO"
  1460 PRINTdble$;name$(1);" IS ";r
ating%;"%"
  1470 PRINTdble$;name$(1);" IS ";r
ating%;"%"
  1480 ENDPROC
```

contributions 23

As you probably know LASERBUG is now willing to pay contributors for their articles. The rate is £5 or £10 per contribution depending on size and content of the article or program.

Articles should be typed or computer printed with double spaces between lines and a margin of at least ¾".

All programs should be sent on cassette with a listing if possible.

All listings should be no wider than 87mm and be formatted using LIST07. If you are using 80 column text use WIDTH34 to get the correct width or WIDTH55 if you are on 132-columns.

Any handwritten material may be subject to delay and error. Please check all programs are bug free.

Higher rates can be negotiated for outstanding work. Please send only original work and make sure you have not sent it elsewhere for possible publication.

Meanwhile, keep writing . . .

classified

Got something to sell or buy? If so then LASERBUG classified ads are for you. Rates are 10p/word (minimum 15 words). Send your ads with cheque/PO made payable to LASERBUG to the normal address. We will get your ad in the next possible magazine.

BBC SOFTWARE
AMATEUR RADIO SOFTWARE for BBC Micro only. RTTY Transceive £7.50, Morse Tutor £3.00, Morse Keyboard £3.00, QTH Locator £3.00, Multifile £6.00, Plus software for running Teleprinters and Teletypes as line printers. Terminal Units (and PCB's) also available. SAE Full lists. (Cheques payable to Mr. R. Sterry) BLT Software, 1 Wavell Garth, Sandal, Wakefield, West Yorkshire WF2 6JP. *Ref: LC001*

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. You will receive your membership card either with this magazine or the next one posted to you. A reminder to re-join will be sent with the penultimate magazine in your subscription.

LASERBUG SPECIAL OFFERS

DUST COVERS FOR YOUR BBC MICRO ONLY £3.25
5% DISCOUNT ON ALL GOODS FROM DRACAL LTD
BACK COPY AND PHOTOCOPY SERVICE
10% DISCOUNT ON ALL BOOKS FROM INTERFACE
25% DISCOUNT ON ALL PROGRAMS FROM FUTURA SOFTWARE
REPLACEMENT MEMBERSHIP CARD SERVICE

LASERBUG Dust Covers are manufactured for us by a well known company and are made to extremely high quality. For full details on the covers see LASERBUG Issue 5. Price £3.25.

Dracal (North West) Ltd. have kindly agreed to give all LASERBUG members a 5% discount on all their goods. These include design sheets and monitor stands. Full details from Dracal (North West) Ltd., PO Box 130, Warrington, WA1 4QB. Phone Padgate 815419. Also see LASERBUG Issue 6. Please quote your membership number with your order. Remember all your dealings are with Dracal, not us. We are not acting as agents for Dracal and cannot be held responsible for their service.

Missed out on a copy of LASERBUG??? If so then you require our back copy service. Back copies are available for the price of £1.25 + postage each and we have back copies of all magazines except Issue 1 (see our photocopy service). Brief details of each magazine are given below.

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 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, Rainbowspot

For those magazines which we have run out of we offer a photocopy service which also applies to the photocopying of individual articles. A photocopy of Issue 1 costs £1.60 plus postage. If there are any articles you wish photocopied this costs 50p/article.

Interface have kindly allowed us to offer you a discount on all their books. However unlike the Dracal offer all your dealings are through us. The two books on offer are Making The Most Of Your BBC Micro and The BBC Micro Revealed. Full details in Issue 8.

Yet another discount for LASERBUG members. Futura Software have allowed all LASERBUG members to have a 25% discount on their software. Futura's current range is: Space Games Pack One (UFO Invasion, Corridor of Lasers, Robot Chase), Space Games Pack Two (Alien Star Fighter, Missile Launch, Zayon Alert), Space Games Pack Three (Asteroid Rider, Doom Satellite, Invaders from Mars), Space Games Pack Four (KVX 153, Space Walk, Delta Wing Attack). All four games are £4.49 (£3.74 with discount). Also available are Beam Me Up Scotty (32k only) which is a Star Trek type game for £6.95 (£5.21 with discount), Star Raider (32k only/Joystick optional) for £5.95 (£4.46) and Powerboat Race (32k only/Joystick required) for £5.95 (£4.46 with discount). All programs are available mail order only. More details from Futura Software, 63 Lady Lane, Chelmsford, Essex, CM2 0TQ. Please quote your membership number with your order to qualify for the discount.

Tatty membership card? For the minor charge of 50p plus your old membership card we will replace it for you. We cannot do this unless we have your old card. We **WILL NOT** replace lost cards.

Please send me ..... LASERBUG Dust Cover(s) @ £3.25 each £..... .....p

..... LASERBUG Back Copies of Issues ..... @ £1.25 each £..... .....p

..... LASERBUG Issue 1(s) (Photocopies only) @ £1.60 each £..... .....p

..... Making The Most Of Your BBC Micro by Tim Hartnell @ £5.81 each £..... .....p

..... The BBC Micro Revealed by Jeremy Ruston @ £7.16 each £..... .....p

..... total number of items. Postage @ 20p/item £..... .....p

TOTAL ENCLOSED £..... .....p

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

.....

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

PLEASE ALLOW 28 DAYS FOR DELIVERY

(This order form supercedes all others)