



**Issue 8 Jan'83**

**IN THIS ISSUE:**

- Memory analyser
- Diskspot
- Focus on adventures
- Computer conversions Part II

..... and lots more!

It's the start of the new year, 1983 with not much news to report. This year will be the "make or break" time for the BBC Computer. Come April it reaches its first birthday and should be established. I say should because when you look at its record in the past year things haven't been going too well. Unless this year Acorn sorts itself out and gets the whole project running smoothly the whole thing is likely to just fade out which would be a great pity. The BBC Micro is an excellent machine and in my (and many other peoples) opinion is the best home computer available in anywhere near the same price range - we both know that otherwise I wouldn't work for LASERBUG and you wouldn't be reading it. Lets hope that Acorn gets itself straight, good software keeps on coming in and people keep on buying the machine.

So what has happened recently. Not a great deal it seems. There are two meetings in the news - one last month and the other the beginning of this. The Computer Programme is sold to America. Logo with a proper turtle is available for the BBC. Torch have put down their prices. A version of FORTH in ROM form is out as is a 128k RAM pack and a fan for your BBC!

First the two meetings. The Institute of Electrical Engineers held their colloquium on "The BBC Computer Literacy Project - Has it succeeded?" on Friday 10th December in which many interesting things were said so our "man-on-the-spot" Dr. Susans reports. Acorn had their first production delivery of OS 1.2 ROMs on the 8th December but from what was said these will be used in new computers. All mail order software from Acornsoft ordered before the 10th was due to arrive before Christmas (any confirmation of that?) however dealers will not be given full deliveries until February. Delivery dates were given for many of the new pieces of hardware (see last months Editorial for details about the Inkjet printer.) The new speech ROM was in action but not very impressive. An improved system will be out in April. Mr. Radcliffe of the BBC said that he was not at all satisfied with the quality of any of the existing BBC Software but the new software being issued in about March would be much better. Also in about March or April the BBC was going to licence a small BBC Computer controlled "buggy" with a 5m cable to the computer so that control systems can be investigated. The other event is a BBC Microcomputer System Trade Exhibition on January 5th. This would have all finished by the time you read this but entrance was by invitation only.

Beating Acorn to launching the BBC Computer in America, the BBC's distributor Films Incorporated have sold the Computer Programme to many different American channels. It will be interesting to find out how the Americans react to the programmes.

A version of Logo has come onto the market aimed primarily at educational establishments. More than just the language this package includes the actual "turtle". The pack costs £350 and consists of the actual turtle, and RS232 interface with connecting cable and of course the necessary software. The system requires the 1.0 OS or higher. Further details from Jessop Electronics, Unit 5, 7 Long Street, London EC2 8HN.

Torch Computers have put down the price of the Torch Z80 Disk Pack from £1144 to £895 making it even more attractive to BBC Micro Business users. Prices of the Torch Computer has also dropped.

Interested in FORTH but can't be bothered to use a tape version or even want to use it as the main language on your computer. If so then HCCS Associates have a FORTH ROM available. It costs just £34.72 + £6.75 for a manual. More details from HCCS Associates, 533 Durham Road, Lowfell, Gateshead, Tyne & Wear NE9 5EY.

Need more memory on your BBC Micro? Micro Management have a 128k RAM pack available which plugs straight in your computer and is addressable for your own programs. Cost - £344.85. Also by the same company is a fan for your BBC Micro to solve those overheating problems for £41.74 inclusive. More details on both these products from Micro Management, 32 Princes Street, Ipswich.

That's it from me this month - a much shorter Editorial than usual. Never mind, I'll make up for it next month.

Happy New Year . . . . .

Please address all correspondence to:

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

It helps us considerably when sorting all the mail if, in the top left hand corner, you can write one or two words that describes the contents of your letter i.e. PROGRAMS, MEMBERSHIPS, QUERIES. No letter can be replied to unless a SAE is enclosed.

Paul Barbour

Dear LASERBUG,

I have just received the November Issue of LASERBUG and was interested to see the article on Software Protection.

Phil Hirst goes to great lengths to explain how to establish the execution address of a m/c program when BBC BASIC provides a direct command. I refer to page 398 of the user guide:

Type \*OPT1,2  
\*LOAD""

When the program has been loaded the screen will display the start address, end address and the execution address thus making the operation to produce back-up copies very easy (using this you will get a message line ASTROCRASH 24 2500 00001C13 00002000 - you should save this by \*SAVE"ASTROCRASH"1C13+2500 2000 - Ed.)

I hope this may be of use to other programmers.

Yours sincerely,  
Alan Mothersole,  
Hillingdon, Middlesex.

*The following letter was originally addressed to Acorn Computers but the writer has asked us to reprint it as an open letter to Acorn as they refuse to reply to him.*

Dear ACORN COMPUTERS,

Exasperation has finally got the better of me and I am now putting onto paper the letter I have been meaning to write for months.

As an owner of a BBC Microcomputer, I am concerned and often annoyed at the way in which your organisational problems are shortening the commercial lifespan of my computer. Many computer dealers in the West Midlands have stopped dealing with Acorn and the brave few that continue are bitter about the lack of support they receive and concentrate on selling Dragon machines.

I resent all the complaints I here about faulty machines. I resent the wait since September for Acornsoft packages and my college library's wait since August for two Acornsoft books. The BBC Micro is an excellent machine but it will not be long before either something better or cheaper or both is available with insufficient BBC Micros already in use to sustain interest.

A great number of people have shown their support for the British computer industry so far and want that industry to succeed, yet that industry seems not to care. Technical problems with hardware may be forgiveable but not so for simple cassette copying and warehouse organisation.

Yours sincerely,  
David King,  
Wolverhampton, W. Midlands.

*continued on page 4*

## help!

Our occasional plea for help goes out to all programmers out there if you use FORTH. We would like someone to review a copy of FORTH for the Software section as nobody on the LASERBUG team is conversant enough in the language.

Please get in touch to our usual address marking the envelope FORTH. Thank-you . . . . .

## contents

● Editorial	2
● Letters	2
● Softreview	3
● Softspot repeat	5
● Memory analyser	7
● Queryspot	8
● TV test signal generator	8
● Screen dumps revisited	12
● Assembler programming on the BBC micro Part I	13
● Rainbowspot	15
● Printreview	17
● Computer conversions Part II	18
● Corrections	18
● Meeting place	18
● Errors?	18
● Focus on adventures	19
● Winter scene - seasonal 4	20
● Competition 3 results	21
● Diskspot	22
● Bookreview	22
● Oddspot	22
● Credits	23

**MISCELLANEOUS PROGRAM:** The Computer Programme Programs  
Volume 1

**REQUIREMENTS:** 16k

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

**PRICE:** £10.00

**REVIEWED BY:** LASERBUG Team

**DESCRIPTION OF PROGRAM:** This tape is a collection of 12 programs that were used in the first series of The Computer Programme. Most of them you will have already seen if not all of them so I will only describe each one briefly. Each program is written in such a way so as to be easily understood and extensive use is made of REMs. The first program simply draws the BBC logo, the Owl. Following on from this is a program that shows the pattern of a ball bouncing and the sales graph program. Next is a bubble sort program which works on a list of names and a 3D cube revolving. After this a remarkably useless program that prints up all (or should I say some) of the many computer languages available. The next program to me looks as if a drunk spider with his feet dipped in ink has walked across the screen but in actual fact the program "plots the elliptical function of three randomly selected values, one of which controls horizontal movement, another vertical movement and the third specifies the number of petals in the flower". An anagram program is next for all puzzle addicts and the program that demonstrated weaving (actually it was about binary numbers) follows. Three programs left. One turns the screen sideways and allows you to use write on it much more like a piece of paper. The penultimate program shows you an example of logic and finally you are given a program that plots 9 different mathematical functions. Who would buy this program I am not really sure. Apart from being nostalgic all the programs are interesting and some are useful but with so much software around for set purposes I am unsure of this packs popularity. Personally I found them all interesting and enjoyed reviewing them.

**PRESENTATION:** ★ ★ ★

**USE OF GRAPHICS:** ★ ★ ★ ★

**VALUE FOR MONEY:** ★ ★ ★

—o0o—

**ARCADE GAME PROGRAM:** The Frog

**REQUIREMENTS:** 32k

**SUPPLIER:** Software for All, 72 North Street, Romford, Essex

**PRICE:** £8.95 (Provisional – might be different by +/- £1)

**REVIEWED BY:** LASERBUG Team

**DESCRIPTION OF PROGRAM:** Generally the cream of the programs written are milked off by Acornsoft. Hence the software produced by the independent software houses is normally second rate when compared to these. In this case however Acornsoft have come unstuck because Software for All have certainly landed a big catch here with this version of Frogger. The Frog is an excellent version of the arcade game with extensive use of sound, graphics and colour making this a very faithful copy of the original. It is the best version I know of out of at least 3 available for the BBC plus several for other computers and TV games.

**PRESENTATION:** ★ ★ ★ ★

**ADDICTIVE QUALITY:** ★ ★ ★ ★ ★

**USE OF GRAPHICS:** ★ ★ ★ ★ ★

**VALUE FOR MONEY:** ★ ★ ★

—o0o—

**GAMES PROGRAMS:** Games Pack 1 (UFO/COL/ROBOT)

**REQUIREMENTS:** 16k

**SUPPLIER:** Futura Software, 63 Lady Lane, Chelmsford, Essex CM2 0TQ

**PRICE:** £4.99

**REVIEWED BY:** LASERBUG Team

**DESCRIPTION OF PROGRAM:** Three very simple games are on this tape. The first, UFO Invasions involves you shooting down flying saucers with a gun that can fire in one of three positions. With Corridor of Lasers you must walk through a corridor with hidden lasers without getting zapped. Finally, Robot Chase is a simple chase game with you running away from a robot whilst trying to find a hidden escape route. The games might sound exciting with those descriptions but although good use is made of user definable characters the games themselves are simple in the extreme and are not worth the effort of buying. I would advise Futura to greatly improve the standard of the games if they expect anyone to be seriously interested in them.

**PRESENTATION:** ★ ★

**ADDICTIVE QUALITY:** ★ ★

**USE OF GRAPHICS:** ★ ★ ★

**VALUE FOR MONEY:** ★ ★

—o0o—

**ARCADE GAME PROGRAM:** Meteors

**REQUIREMENTS:** 32k (and 6522 VIA chip probably – Ed.)

**SUPPLIER:** Acornsoft, 4a Market Hill, Cambridge CB2 3NJ

**PRICE:** £9.95

**REVIEWED BY:** Trevor Lawford

**DESCRIPTION OF PROGRAM:** This excellent arcade-game program is a very good copy of "Asteroids" (don't let Atari hear you say that – Ed.). The outline graphics and movement are far superior to the Atari cartridge and the MODE 1 resolution and colours are well up to the standard of the original. The game has been written by the same author as that of "Defender" (or "Planetoid" on the later versions). If Defender is the best game for the Beeb then "Meteors" comes a good second. Acornsoft really know how to produce games that are streets ahead of the rest. Highly recommended. Also unusual for a home computer, it features a "display mode" between games. Ship handling is superb!

**PRESENTATION:** ★ ★ ★ ★ ★

**ADDICTIVE QUALITY:** ★ ★ ★ ★

**USE OF GRAPHICS:** ★ ★ ★ ★ ★

**VALUE FOR MONEY:** ★ ★ ★

—o0o—

**ARCADE GAME PROGRAM:** Cassette 9 – Model B Invaders

**REQUIREMENTS:** 32k

**SUPPLIER:** IJK Software, 55 Fitzroy Road, Bispham, Blackpool, Lancs.

**PRICE:** £6.95

**REVIEWED BY:** LASERBUG Team

**DESCRIPTION OF PROGRAM:** This game is the best version of Space Invaders that I have seen so far. Although the graphics are different the game follows closely that of the original. Several options are available including invisible invaders. A really good game – if you want an invaders program that plays the classic version, this is the one for you.

**PRESENTATION:** ★ ★ ★ ★

**ADDICTIVE QUALITY:** ★ ★ ★ ★

**USE OF GRAPHICS:** ★ ★ ★

**VALUE FOR MONEY:** ★ ★ ★

—o0o—

**ADVENTURE PROGRAM:** Philosopher's Quest

**REQUIREMENTS:** 32k

**SUPPLIER:** Acornsoft, 4a Market Hill, Cambridge CB2 3NJ

**PRICE:** £9.95

**REVIEWED BY:** Doris Ratley

**DESCRIPTION OF PROGRAM:** This program loads in three parts the last of which needs re-loading every time you "die" or quit the program. The program needs a lot of thought as it is easier to be horribly killed, e.g. eaten by spiders, falling and breaking every bone in your body or being crushed to death, than it is find treasure. In half an hour all I managed to score was 14 points maybe I should take advantage of the card supplied with the game and ask Acorn for a clue.

**PRESENTATION:** ★ ★

**COMPLEXITY:** ★ ★ ★ ★

**RESPONSE SPEED:** ★ ★

**VALUE FOR MONEY:** ★ ★ ★

—o0o—

**EDUCATIONAL PROGRAM:** Metrics

**REQUIREMENTS:** 32k

**SUPPLIER:** Chalksoft, Lowmoor Cottage, Tonedale, Wellington, Somerset TA2 10AL

**PRICE:** £9.95

**REVIEWED BY:** LASERBUG Team with help from children/teachers

**DESCRIPTION OF PROGRAM:** This tape consists of five separate programs, each one being a multi-choice test on mass, capacity, length, area and volume respectively. Each multi-choice starts off with simple questions which get more and more complicated finishing off with a few sums to do. The programs were good and well presented and the class enjoyed trying the programs to reinforce work already taught.

**PRESENTATION:** ★ ★ ★

**FOR AGES:** 10–16

**SUBJECT:** Maths – the Metric System

**USEFULNESS:** ★ ★ ★ (Revision)

**NUMBER OF USERS:** 1

**VALUE FOR MONEY:** ★ ★ ★

—o0o—

**EDUCATIONAL PROGRAM:** Multiply & Divide

**REQUIREMENTS:** 16k

**SUPPLIER:** Cottage Software, Heather Cottage, Selly Hill, Whitby, N. Yorkshire

**PRICE:** £7.50

**REVIEWED BY:** LASERBUG Team with help from children/teachers

**DESCRIPTION OF PROGRAM:** These programs which test the children on long multiplication and division are comparable I feel with the style of programs on Early Learning. They take the child step through step with the

questions with the child being able to answer either easy or hard ones with 2 or 5 questions. If the child manages to answer all the questions without an error he gets to play a game. On multiply this is 6 games of noughts and crosses and on divide it is a game of hangman where you have to guess a mathematicians name – if you get it right you are told what he is famous for adding a bit of history to the program as well! A good program and well worth buying.

**PRESENTATION:** ★ ★ ★

**FOR AGES:** 10–12 (or anyone of suitable ability)

**SUBJECT:** Maths (long multiplication and long division)

**USEFULNESS:** ★ ★ ★

**NUMBER OF USERS:** 1

**VALUE FOR MONEY:** ★ ★ ★

—o0o—

We would like to thank BBC Publications, Software for All, Futura Software, Acornsoft, IJK Software, Chalksoft and Cottage Software for supplying us with copies of their programs to review. Thanks also are due to Trevor Lawford and Doris Ratley for sending us their reviews of programs.

As a new feature we would like to offer our top 8 programs of the month. These are based entirely on the opinions of the LASERBUG Team and people we have talked to at the moment. If you would like to send your votes in you are welcome. Also software houses are of course free to send us software which we will consider for inclusion in the charts.

1st	Arcadians	Acornsoft
2nd	Defender/Planetoid	Acornsoft
3rd	Monsters	Acornsoft
4th	The Frog	Software for All
5th	Meteors	Acornsoft
6th	Invaders	IJK Software
7th	Zombies	Software for All
8th	Martians	Program Power

If sufficient interest is generated we would be prepared to print your high scores in LASERBUG – let us know.

## BBC FORTH BBC FORTH

Level 9 Computing are pleased to announce a new compiler for the increasingly popular language FORTH on BBC A & B micros.

FORTH is a powerful, extensible language, simple in concept & use, that encourages structured programming and is good both for large programs and simple one-off utilities.

"r q FORTH" is supplied on cassette, with a 70 page technical manual and a summary card, for £15 including VAT/P&P. It:

- \* runs up to 10 times faster than BBC BASIC;
- \* includes a full screen editor, tailored for the BBC;
- \* is FORTH-79 STANDARD and provides fig-FORTH facilities so it is simple to use programs published in either dialect;
- \* provides 260 FORTH words (i.e functions) initially;
- \* is readily extensible (even defining words can be defined);
- \* allows full use of the M.O.S facilities from within FORTH;
- \* allows use of all graphic modes, even 0-2 (just!);
- \* provides recursion simply and naturally;
- \* needs no added hardware and will run on an unexpanded 'A';
- \* is available NOW.

## BBC adventures

**Colossal Adventure:** The classic mainframe game "Adventure" with all the original treasures & creatures & 70 extra rooms.

**Adventure Quest:** Through forest, desert, mountains, caves, water, fire, moorland and swamp on an epic quest vs Tyranny.

**Dungeon Adventure:** NEW! The dungeons of the Demon Lord have survived His fall. Can you get to their treasures first?

Every Level 9 adventure is packed with puzzles and has over 200 individually described locations – a game can easily take weeks to complete! Only data and code compression allow so much to be provided.

Each adventure needs 32K and costs £9.90, including P&P/VAT.

Send order or SAE for catalogue, describing your computer, to

**LEVEL 9 COMPUTING**

Dept L, 229 Hughenden Road, High Wycombe, Bucks. HP13 5PG

continued from page 2

Dear LASERBUG,

Having just read my November issue of LASERBUG I must comment on the results of the competition presented. The program that was written to be the fastest is not the fastest. If you insert \*FX16,0 at the start of the program just after the initial linenumber you will gain another 2/100 sec. This can be set back to \*FX16,4 when the program ends.

\*FX16,X is a call controlling the number of A/D convertors being read. Setting this to zero (X) disengages this reading and all BASIC programs run a little faster.

Furthermore, when it comes to the competitions I can't compete since my issues reach me about the middle of the month FOLLOWING the month of the issue. Could you speed this up.

Yours sincerely,  
Janne Soderberg,  
Jarfalla, Sweden.

(As long as your answer to the competition is postmarked by the closing date we will still consider it – Ed.)

Dear LASERBUG,

First and foremost, thanks Paul for your individual effort over the months. The amount of programs and articles by you alone should put the rest of us to shame. Just one small point though: Seasonal articles do tend to show the lateness of the newsletter i.e. I received my November issue with the firework program in during the first week of December but perhaps I'll bash it in ready for next year.

A useful key function when typing letters if you can't be bothered to load a wordprocessor – try \*KEY0""",MPRINT""", then use AUTO line numbering but instead of using RETURN at the end of the line use f0.

Yours sincerely,  
John Murphy,  
Telford.

Dear LASERBUG,

I write with a plea for help. Five times I have returned a tape for Bugbyte Computer Chess for the BBC Micro. Each time I ask if they will please send one which has been tested but each time it either will not load or loads but prints 'bad program' when I try to run it. I have access to three other sources for testing the tape on a BBC machine and so I am certain that the fault is not a loading one on my part. I first ordered BBC Chess at the beginning of August '82 and by the beginning of September my ACCESS account showed that Bugbyte had already charged me for it even though I did not receive my first tape until well into September. Since then I have been sending back tapes with pleading letters but all I ever receive is another tape on its own in an envelope. At one stage they did include another game in the envelope but to my disgust it would not load either.

Would you please try to find out from other BBC owners if they have had the same problems with BUGBYTE because the glossy advertisement in the computer magazines gives the impression that it is a very professional company which in my experience it certainly is not.

The first 'Chess' tape which I had trouble with was verified as a 'bad program' by the centre for Scottish Micro-Computer Education which have around ten BBC machines.

Yours sincerely,  
John Shaw,  
Ruthrglen, Glasgow.

(Try using \*RUN instead of CHAIN""". Other than that readers letters are always welcome – Ed.)

Dear LASERBUG,

Many thanks to you for saving me £120!

I have had my BBC Micro since February (give or take ten weeks while it was away being repaired twice, getting lost once and finally being replaced with another one) and I was conscious of under-using its potential as I only had a black and white portable TV as a display. However I was reluctant to spend £300 to £350 on a colour monitor which would not receive TV, or on a modified receiver with uncertain servicing facilities, knowing that an equivalent (or better) receiver would be less than £200 from a local branch of a discount chain. Luckily I had not committed myself in resolving this conflict before seeing Dr. Susans' article in the October issue of the magazine.

After reading this article and corresponding briefly with Dr. Susans I took the plunge and bought a Ferguson 14" portable colour TV (model 3755) costing £179. This has a TX9 chassis which is slightly different from the TX10 referred to by Dr. Susans but my elementary knowledge of electronics, some modifications to his resistor values, £2.48 worth of components specially bought and a raid on my workshop for the rest produced exactly the desired result. I now have a switchable receiver/monitor which can be easily serviced locally in case of any trouble (although the guarantee may not be valid) but which cost me in total only £182 instead of £299 for the unknown factor commercially available equipment.

Admittedly not everyone would have a suitable mains isolating transformer lying about, but even allowing £30 to buy one of these the

modification is still a bargain to my mind.

The resolution of even a modified TV is not as good as a pure monitor of course but I find MODE0 perfectly readable with a clean colour background, unaffected by the patterning which makes it a strain to use this mode on an ordinary colour TV.

I'm not sure whether it is an asset or not but I can also listen to the TV sound if I wish, even when using the set as a monitor.

The sharing of information, which often means the "knowledgeable" being prepared to pass on their knowledge to the benefit of the rest of us, is exactly what a good user group is all about. Keep it up!

Yours sincerely,  
David Prideaux,  
Crownhill, Plymouth.



BRITISH BROADCASTING CORPORATION  
TELEVISION CENTRE WOOD LANE LONDON W12 7RJ  
TELEPHONE 01-743 8000 TELEX 265781  
TELEGRAMS AND CABLES TELECASTS LONDON TELEX



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

c/o Room 501 E. Tower  
DEC 27 1982

Dear Paul Barbour,

Would you be kind enough to mention in Laserbug that I'm putting together a third series of ADVENTURE GAME. This time a Beeb micro will have found its way to the planet Arg (replacing the Apple) and I'm looking for ideas for its use to put in the twisted minds of the Argonds! Those who have seen the series will know what I mean; those who have'nt... well its really too lengthy to explain.

Problems need to be not too simple but have a quick solution once the key point is discovered. The solution should also be available to the viewers of course. 'Arcade-type' games are not really on, they tend to be long-winded and be more fun to play than to watch.

I'll be grateful for any ideas from your members. No prizes I'm afraid but I'll try and acknowledge everything sent in. Address as above.

Yours sincerely

PATRICK DOWLING (Producer 'Adventure Game')

## softspot – repeat!

Our games program this month is a computerised version of the Simon game that most of you will be familiar with. The program is only suitable for a Model B (or 32k A) and will not take too much altering as it uses up most of the available memory. Disk users will have to reset PAGE to &EOO for the game to work.

Paul Barbour

>LIST

```

10 REM REPEAT
20 REM by Paul Barbour
30 :
40 REM 15/12/82
50 :
60 REM Version 1.0
70 :
80 REM Takes up ~3.75k memory
90 REM (~11.6k with variables)
100 REM and uses MODEs 2 and 7
    
```

```

110 :
120 REM Requires 32k
130 :
140 REM Written on OS 0.1 EPROM
150 :
160 REM (C) LASERBUG 1983
170 :
180 : : : :
190 :
200 PROCdv
210 GOSUB390
220 PROCt
230 PROC# : PROC#
240 IFF%=TRUE THEN300
250 PROC# : PROC#
260 GOSUB400
270 PROCsb : PROC#
280 GOSUB390
290 PROC# : PROC#
300 PROC# : PROC#
310 PROC# : PROCsu9
320 GOSUB400
330 PROCsb
340 PROCs
350 PROCr
360 P%=P%+1 : GOTO330
370 : : : :
380 :
390 MODE7 : VDU23 : 8202 : 0 : 0 : 0 : : RETU
RN
400 MODE2 : VDU23 : 8202 : 0 : 0 : 0 : : RETU
RN
410 :
420 DEFPROCdv
430 r#=CHR#129 : b#=CHR#132 : c#=CHR
#134 : g#=CHR#130 : w#=CHR#135 : y#=CHR#
131 : wb#=" "+CHR#157 : yb#=CHR#131+CH
R#157 : swb#=CHR#157 : bb#=CHR#156 : d#=
CHR#141
440 ENVELOPE1,3,0,0,0,0,0,121,
-10,-5,-2,120,120 : ENDPROC
450 :
460 DEFPROCt
470 PRINTTAB(16,8) : d# : "REPEAT" : T
AB(16) : d# : "REPEAT" : TAB(12,15) : d# : "
by Paul Barbour" : TAB(12) : d# : "by Pa
ul Barbour" : *FX15,0
480 AX=INKEY(500) : ENDPROC
490 :
500 DEFPROC#
510 CLS : PRINTwb#wb# : TAB(15) : r# :
d# : "REPEAT" : wb# : TAB(15) : r# : d# : "REP
EAT" : wb# : yb# : TAB(11) : b# : "(C) LASERB
UG 1982" : : ENDPROC
520 :
530 DEFPROC#
540 PRINTc# : "Do you want instruc
tions ("Y" or "N") ?" : REPEATUN
TILINKEY(-86)ORINKEY(-69) : IFINKEY(
-86)THENF%=TRUE : ELSEF%=FALSE
    
```

```

550 ENDPROC
560 :
570 DEFPROC i
580 PRINT c$; "    The idea of th
is game is for you" c$; "to copy a
sequence that the computer" c$; "wi
ll give.  Each time you repeat a"
c$; "sequence correctly another "e
lement" "" c$; "is added.  Each "e
lement"" is a colour"
590 PRINT c$; "with an associated
sound - there are" c$; "four "elem
ents"" in all.": TIME=0: REPEAT UNTIL
TIME=500: PRINT "" g$; " Press the"
; w$; s w b$; b$; "SPACE BAR "; g$; b b$; "t
o see the"
600 PRINT TAB(12); g$; ""elements"
"...": REPEAT UNTIL INKEY(-99): ENDPRO
C
610 :
620 DEFPROC sb
630 CLS: FOR col% = 1 TO 4: VDU 19, col%
; 0; 0: NEXT: GCOL 0, 1: MOVE 0, 0: DRAW 640,
0: PLOT 85, 640, 512: DRAW 0, 512: PLOT 85,
0, 0: GCOL 0, 2: MOVE 640, 0: DRAW 1280, 0: P
LOT 85, 1280, 512: DRAW 640, 512: PLOT 85,
640, 0: GCOL 0, 3: MOVE 640, 512: DRAW 1280
, 512: PLOT 85, 1280, 1024: DRAW 640, 1024
640 PLOT 85, 640, 512: GCOL 0, 4: MOVE 0
, 512: DRAW 640, 512: PLOT 85, 640, 1024: D
RAW 0, 1024: PLOT 85, 0, 512: ENDPROC
650 :
660 DEFPROC d
670 level% = 25: TIME = 0: REPEAT UNTIL
TIME = 100: PROC f(4, 1): PROC d e: PROC f(3
, 3): PROC d e: PROC f(1, 6): PROC d e: PROC f
(2, 4): PROC d e: TIME = 0: REPEAT UNTIL TIM
E = 200: ENDPROC
680 :
690 DEFPROC f(col1%, col2%)
700 VDU 19, col1%, col2%, 0; 0: SOUND
&11, 1, 5 + (48 * (5 - col1%)), level%: SOUN
D &12, -15, 5 + (48 * (5 - col1%)), 1: TIME = 0
: REPEAT UNTIL TIME = level%: VDU 19, col1
%; 0; 0: ENDPROC
710 :
720 DEFPROC d e
730 TIME = 0: REPEAT UNTIL TIME = 50: EN
DPROC
740 :
750 DEFPROC i2
760 PRINT c$; "    As you saw the
re are four colours:" c$; "Red, Ye
llow, Cyan and Blue." y$; "="; TAB(
7); "="; TAB(16); "="; TAB(25); "=" c$;
"These are controlled by their ini
tial" c$; "letters on the keyboard:
"
770 PRINT "" TAB(14) w$; "W E"; r$; "R
"; w$; "T"; g$; "Y"; w$; "U" TAB(14) w$; "
S D F G H J" TAB(14) w$; "X"; c$; "C";
w$; "V"; b$; "B"; w$; "N M" "" c$; "The c
omputer will give you the sequence
";
780 PRINT c$; "and you must repeat
it using the RYCS" c$; "keys.": TIM
E = 0: REPEAT UNTIL TIME = 500: PRINT "" g$;
"Press the"; w$; s w b$; b$; "SPACE BAR
"; g$; b b$; "to continue...": REPEAT UN
TIL INKEY(-99): ENDPROC
790 :
800 DEFPROC i
810 PRINT c$; "What skill level (1
-50, 50 easy/1" c$; "impossible!)"
c$;
820 *FX 15, 0
830 INPUT level%: IF level% <= 0 OR lev
el% > 50 THEN PRINT TAB(1, 8); SPC(80); : P
RINT TAB(1, 8); : GOTO 820
840 level% = level% + 5: ENDPROC
850 :
860 DEFPROC su e
870 PRINT c$; d$; " Please wait whi
le I set up the game" c$; d$; " Plea
se wait while I set up the game": a
bsolute_sequence$ = "" : REPEAT: absolu
te_sequence$ = absolute_sequence$ + ST
R$(RND(4)): UNTIL LEN(absolute_seque
nce$) = 255: P% = 1
880 ENDPROC
890 :
900 DEFPROC s
910 sequence$ = LEFT$(absolute_seq
uence$, P%): FOR sequence% = 1 TO LEN(seq
uence$): flash% = VAL(MID$(sequence$,
sequence%, 1)): IF flash% = 1 THEN PROC f(
4, 1)
920 IF flash% = 2 THEN PROC f(3, 3)
930 IF flash% = 3 THEN PROC f(1, 6)
940 IF flash% = 4 THEN PROC f(2, 4)
950 NEXT: ENDPROC
960 :
970 DEFPROC r
980 COLOUR 6: PRINT TAB(3, 15); "NOW
REPEAT THE"; TAB(6, 17); "SEQUENCE": T
IME = 0: REPEAT UNTIL TIME = 100: PROC sb: s
ub_pointer% = 1: REPEAT: element% = 0: *F
X 15, 0
990 element$ = INKEY$(level% * 20)
1000 IF element$ = "" THEN 1120
1010 IF element$ = "R" THEN element%
= 1
1020 IF element$ = "Y" THEN element%
= 2
1030 IF element$ = "C" THEN element%
= 3
1040 IF element$ = "B" THEN element%
= 4
1050 IF element% = 0 THEN 1120
1060 IF element% = 1 THEN PROC f(4, 1)

```



```

500  IFAZ=&89 THEN XZ=XZ+1:IFYZ
=40 THENXZ=0:YZ=YZ+1:IFYZ=24 THEN
YZ=23:PROCuP
510  IFAZ=&8A THEN YZ=YZ+1:IFYZ
=24 THEN YZ=23:PROCuP
520  IFAZ=&8B THEN YZ=YZ-1:IFYZ
=-1 THEN YZ=0:PROCDowN
530  UNTILFALSE
540  :
550  :
560  :
570  DEFPROCdisplay
580  PRINTCHR$(30);CHR$( &83);"Add
ress:";PZ+XZ+YZ*40;"/&";~PZ+XZ+YZ*
40;
590  PRINT" Contents: ";?(PZ+XZ+Y
Z*40);"/&";~?(PZ+XZ+YZ*40);
600  IFPOS<39 THEN PRINT STRING$(
39-POS," ")
610  ENDPROC
620  :
630  DEFPROCuP
640  VDU28,0,24,39,1
650  VDU31,0,24,10
660  VDU26
670  PZ=PZ+40
680  FORTZ=0TO39
690  AZ=?(TZ+PZ+23*40)MOD 128
700  IFAZ<32 OR AZ>126 THEN AZ=
0
710  ?(HIMEM+TZ+24*40)=AZ
720  NEXTTZ
730  ENDPROC
740  :
750  DEFPROCDowN
760  VDU28,0,24,39,1
770  VDU30,11
780  VDU26
790  PZ=PZ-40
800  FORTZ=0TO39
810  AZ=TZ?PZ MOD128
820  IF AZ<32 OR AZ>126 THENAZ=
0
830  ?(HIMEM+TZ+40)=AZ
840  NEXTTZ
850  ENDPROC
860  :
870  REM Error
880  *TV
890  MODE7
900  *FX4
910  REPORT
920  PRINT" at line "JERL

```

## queryspot

**Q. Just thought I would write and tell you how much I enjoyed reading the first four issues of LASERBUG. Keep up the good work!!!**

You will probably be surprised to learn that you have a member over here (not really, we have members in 15 different countries – Ed.) and even more surprised to learn that my Model 'B' arrived only 4 months after I ordered it. However nothing comes free as the power supply packed up two hours after I turned the machine on. After a phone call to Acorn, they promised to send a new one which took two months to arrive (surface mail!). I'm now back in business again.

Which brings me to the point of this letter. I have one other small problem, which is rather irritating. I wrote to Acorn about it and they replied with a standard PR letter. I was wondering if you could mention it at the next get-together, and possibly to our man at Acorn to see what they think.

When changing MODEs in the middle of programs, or going to MODE5 from 7, I notice random flashes of coloured lines, or a repeat of the last displayed screen. An example should clarify this. If I go from MODE7 to 5, I get a split second flash of slightly bent yellow vertical lines from the bottom of the screen to midway, and then the clear screen. Acorn replied that this was quite normal.

It seems to me that this is not normal, but have no idea if this is another OS 0.1 bug or a hardware problem (ULA?). I should mention that when the power supply went wonky, it was putting out between 5.6 and 5.9 volts.

Any thoughts or suggestions you might have on the above would be most welcome and I look forward to your reply.

Ian Fildes,

Vancouver, British Columbia, Canada.

A. Sorry to disappoint you Ian but this "problem" is quite normal and occurs on all BBC Micros. The reason this happens is uncertain and readers are welcome to correct me if they know different but my guess is that the problem is software based. Say you switch on the computer and turn to MODE 2. After using this you go back to MODE 7. You switched from using 1k of graphics RAM to 20k and back to 1k. Now when you go back up to MODE 2 you start off by using the old 20k which contains the old screen data. Then the computer clears the old data and hence you are left with a blank screen. One way around the problem might be to clear the necessary graphics RAM before you change MODEs.

**Q. I am distressed by Acorn/BBC's refusal to upgrade the unsatisfactory OS 0.1. Is LASERBUG trying to do anything about this? I feel that 0.1 does not comply with the Technical Spec. on which I decided to buy the machine.**

Mr. Foot,

Ashby de la Zouch, Leics.

A. Well Mr. Foot, there are two ways of looking at this. The first which is the way Acorn think of the situation is that you do not need the 1.0+ until you upgrade to disks or teletext, etc. which require the 1.0 to work. They seem to be very set in their ways about this logic. The other way to look at it, which is how you do, is that the 0.1 does not perform up to the Technical Spec. on which you decided to buy the machine. Perhaps you would like to write in again, or indeed any member can do so, and state exact reasons WHY the OS 0.1 does not perform up to its specification. It is all very well saying that something doesn't perform how it should but EXACTLY what doesn't it do that it should according to the specification (BBC Microcomputer System Technical Specification – Issue 3, September 1981).

## tv test signal generator II

Back in Issue 6 we published a TV Test Signal Generator by Phil Hirst. Below is another type of Test Signal Generator which although has poorer presentation than the first, has many more test facilities.

When colour TV receivers or monitors are used to display text rather than moving pictures, any errors in the alignment of the display are much more obvious.

There are a number of well known instruments which have been developed to enable TV receivers (and monitors) to be easily checked and set-up. These test instruments (often called crosshatch generators) are quite expensive, but most of their functions can easily be implemented on a BBC computer. The accompanying program (for Model B or memory expanded Model A) gives all of the commonly used waveforms with additional features enabling the colours to be changed at will. This saves much swapping of internal links in the receiver during alignment.

Unfortunately it is not possible to give alignment instructions here as they differ very considerably from receiver to receiver and trying adjustments in the wrong order can easily lead to the display becoming very much worse. It is essential to refer to the appropriate service manual for that receiver.

The different patterns are displayed when the appropriate keys are pressed



on the keyboard. The key pressed is decoded (lines 160 to 500) and the appropriate procedure or VDU statement called. The "RETURN" key calls the "menu" (520,780) to give a list of key/patterns. The only exception (due to lack of space) is the "space bar" which clears the screen. This is used to set-up the black levels of the three colours.

The foreground (capitals) and background (small letters) colours can be changed on any of the patterns except for colour bars and stripes.

The first pattern "Screen one colour", PROC1c (1250,1320) is used when adjusting the colour purity of the display, it initially gives a red screen and this can then be changed to green or blue.

The next procedure, PROCxhatch (790,890), gives a pattern of horizontal and vertical lines which enable the beam convergence to be checked over the full screen. This display again starts with a red display and would normally be followed by yellow and then white.

The PROCdot (900,960) procedure is an alternative to the previous display and is used in the same way.

The PROCbar (970,1040) procedure is used to check the low-frequency performance of the system i.e. 20kHz to 0.5MHz.

PROCbuzz (1050,1140) is a procedure which is used elsewhere in the program but which can be used in its own right if there is an accompanying TV sound channel as a picture on sound buzz check.

PROCT (1640,1750) is a procedure giving a 1-t pulse and bar which is used for transient testing. Due to the limitations of the computer the pulse is slightly too narrow (0.8-t) and is generated as a rectangular pulse rather than the correctly filtered pulse, however this will not detract unduly from its usefulness. There will be very few television receivers which will display the pulse but it should be shown on RGB displays. A more suitable test for TV receivers is the PROC2t (1760,1850) procedure which generates a 2-t pulse and bar together with a 10-t colour burst. This colour burst is at about 15° away from the ideal angle but is quite suitable for phase checks.

The procedure PROCcbar (1440,1530) generates standard colour bars whilst PROCstripe (1540,1630) generates horizontal colour bars.

The frequency response of the system can be checked using the PROCfreq procedure. Frequencies of 8 MHz. to 0.5 MHz. are displayed in a multiburst format. The frequency bars are of course rectangular pulses rather than the correct sinusoidal waveforms but this difference is not serious in this application. The 5.3 MHz burst has a complex waveform designed to give the correct brilliance on the screen and the correct amplitude when observed on an oscilloscope. A suitable simple waveform cannot be generated for a frequency in this range by this computer without external equipment. On most receivers, it will not be possible to display any frequencies above 4 MHz. and even this frequency will not be displayed on many colour sets. Most RGB displays should resolve the 5.3 MHz. burst but the screen dot pattern prevents all but special high resolution screens showing the 8 MHz. burst. This however should be resolved on a good monochrome monitor.

The castellation procedure PROCcastel (1150,1240) is a quite severe test of EHT regulation and sync. pulling. There are six horizontal half-line length bars at the righthand side of the picture with narrow vertical stripes down each side of the picture. These stripes should be straight. If not (almost always) the lefthand stripe gives an indication of sync. pulling whilst the difference in straightness of the two stripes is an indication of EHT regulation.

The procedures PROCcircle (2040,2270) and PROCstar (1860,2030) are used as a test of the PAL system. A RGB system will show them as white but a PAL receiver will show cross-colour effects. Both of these procedures use sines and cosines at regular intervals and it is of interest to note that, in order to save time, these are stored in an array which is filled at the start of the program (50,110). Also that only one sin and cos is generated by a function, the rest are generated using the expansions:

$$-\sin(A+B) = \sin(A)\cos(B) - \cos(A)\sin(B)$$

$$\cos(A+B) = \cos(A)\cos(B) - \sin(A)\sin(B)$$

The final display is produced by PROCstreak and has its main applications in testing special amplifiers used in TV picture generating equipment.

D.E. Susans

LIST

```

1 REM TV TEST SIGNAL GENERATOR
II
2 REM      by Dr. D. E. Susans
3 :
4 REM      16/12/82
5 :
6 REM      Version 1.1
7 :
8 REM      Takes up ~4.47k memory
9 REM      (~9.94k with variables
    
```

```

)
10 REM      and uses all MODEs exce
PC
11 REM      MODE 3
12 :
13 REM      Requires 32k
14 :
15 REM      (c) LASERBUG 1983
16 :
17 :
18 :
20 MODE7
30 PRINTTAB(8,6);"TV ALIGNMENT
PATTERNS";" "SPC(13);"Model B only"
40 DIMC(180),S(180)
50 S(0)=0:C(0)=1
60 A=SINRAD(2)
70 B=COSRAD(2)
80 FORA%=1TO180
90   S(A%)=A*S(A%-1)+B*S(A%-1)
100  C(A%)=B*C(A%-1)-A*S(A%-1)
110  NEXT
120 MODE6
130 VDU20
140 PROCmenu
150 ?875=2
160 A#=INKEY$(100)
170 IFA#=""GOTO160
180 IFA#=CHR$(13)MODE6:PROCmenu
190 IFA#=""CLS
200 IFA#="R"VDU19,1,1,0;0
210 IFA#="G"VDU19,1,2,0;0
220 IFA#="B"VDU19,1,4,0;0
230 IFA#="C"VDU19,1,6,0;0
240 IFA#="Y"VDU19,1,3,0;0
250 IFA#="M"VDU19,1,5,0;0
260 IFA#="W"VDU19,1,7,0;0
270 IFA#="K"VDU19,1,0,0;0
280 IFA#="r"VDU19,0,1,0;0
290 IFA#="e"VDU19,0,2,0;0
300 IFA#="b"VDU19,0,4,0;0
310 IFA#="c"VDU19,0,6,0;0
320 IFA#="g"VDU19,0,3,0;0
330 IFA#="m"VDU19,0,5,0;0
340 IFA#="w"VDU19,0,7,0;0
350 IFA#="k"VDU19,0,0,0;0
360 IFA#="N"MODE5:PROC1c
370 IFA#="X"MODE5:PROCxhatch
380 IFA#="D"MODE5:PROCdot
390 IFA#="H"MODE5:PROCbar
400 IFA#="Z"MODE5:PROCbuzz(0)
410 IFA#="1"MODE0:VDU20:PROCT
420 IFA#="2"MODE1:PROC2t
430 IFA#="V"MODE2:PROCcbar
440 IFA#="S"MODE2:PROCcstripe
450 IFA#="F"MODE0:PROCfreq
460 IFA#="O"MODE0:PROCcircle
470 IFA#="L"MODE5:PROCcastel
480 IFA#="*"MODE0:PROCstar
490 IFA#="T"MODE4:PROCstreak
    
```

```

500 GOTO160
510
520 DEFPROCmenu
530 VDU23;11;0;0;0,19,0,4,0;0
540 PRINT "Foreground      Colou
r      Background"
550 PRINTTAB(4)"R";SPC(12);"Red"
;SPC(13);"r"
560 PRINTTAB(4)"G";SPC(12);"Gree
n";SPC(13);"g"
570 PRINTTAB(4)"B";SPC(12);"Blue
";SPC(12);"b"
580 PRINTTAB(4)"C";SPC(12);"Cyan
";SPC(12);"c"
590 PRINTTAB(4)"Y";SPC(12);"Yell
ow";SPC(10);"y"
600 PRINTTAB(4)"M";SPC(12);"Magne
ta";SPC(9);"m"
610 PRINTTAB(4)"W";SPC(12);"Whit
e";SPC(11);"w"
620 PRINTTAB(4)"K";SPC(12);"Blac
k";SPC(11);"k"
630 PRINTTAB(4)"N";SPC(12);"Scre
en one colour"
640 PRINTTAB(4)"X";SPC(12);"Cros
shatch"
650 PRINTTAB(4)"D";SPC(12);"Dot
Pattern"
660 PRINTTAB(4)"H";SPC(12);"Hori
zontal bar"
670 PRINTTAB(4)"Z";SPC(12);"Buzz
Pattern"
680 PRINTTAB(4)"1";SPC(12);"1T-P
ulse and bar"
690 PRINTTAB(4)"2";SPC(12);"2T-P
ulse and bar"
700 PRINTTAB(4)"V";SPC(12);"Colo
ur bar"
710 PRINTTAB(4)"S";SPC(12);"Colo
ur stripes"
720 PRINTTAB(4)"F";SPC(12);"Freq
uency sweep"
730 PRINTTAB(4)"O";SPC(12);"Circ
les"
740 PRINTTAB(4)"T";SPC(12);"Stre
ak Pattern"
750 PRINTTAB(4)"L";SPC(12);"Cast
leations"
760 PRINTTAB(4)"*";SPC(12);"Star
burst"
770 PRINTTAB(4)"RETURN";SPC(7);"
Menu"
780 ENDPROC
790 DEFPROCxhatch
800 GCOL0,1
810 FORA%=0TO1203STEP102
820   MOVE0,A%
830   DRAW1279,A%
840   NEXT
850 FORA%=0TO1279STEP106
860   MOVEA%,0
870   DRAWA%,1023
880   NEXT
890 ENDPROC
900 DEFPROCdot
910 GCOL0,1
920 FORA%=0TO1279STEP106
930   FORB%=0TO1023STEP102
940     PLOT69,A%,B%
950     NEXT B%
960 ENDPROC
970 DEFPROCbar
980 VDU19,1,7,0;0
990 GCOL0,1
1000 MOVE0,512
1010 DRAW0,1023
1020 PLOT85,1279,512
1030 PLOT85,1279,1023
1040 ENDPROC
1050 DEFPROCbuzz(Z%)
1060 VDU19,1,7,0;0
1070 GCOL0,1
1080 FORA%=0TO1000STEP185
1090   MOVEZ%,A%
1100   DRAWZ%,A%+93
1110   PLOT85,1279,A%
1120   PLOT85,1279,A%+93
1130   NEXT
1140 ENDPROC
1150 DEFPROCcastel
1160 VDU19,1,7,0;0
1170 GCOL0,1
1180 MOVE0,0
1190 DRAW0,1023
1200 PROCbuzz(512)
1210 GCOL3,1
1220 MOVE1250,0
1230 DRAW1250,1023
1240 ENDPROC
1250 DEFPROCic
1260 VDU20
1270 GCOL0,1
1280 MOVE0,0
1290 DRAW0,1023
1300 PLOT85,1279,0
1310 PLOT85,1279,1023
1320 ENDPROC
1330 DEFPROCvdu
1340 VDU20
1350 VDU19,8,7,0;0
1360 VDU19,9,3,0;0
1370 VDU19,10,6,0;0
1380 VDU19,11,2,0;0
1390 VDU19,12,5,0;0
1400 VDU19,13,1,0;0
1410 VDU19,14,4,0;0
1420 VDU19,15,0,0;0
1430 ENDPROC
1440 DEFPROCcbar
1450 PROCvdu

```

```

1460 FORA%=0T07
1470   GCOL0,8+A%
1480   MOVEA%*171,0
1490   DRAWA%*171,1023
1500   PLOT85,170+A%*171,0
1510   PLOT85,170+A%*171,1023
1520   NEXT
1530 ENDPROC
1540 DEFPROCcstripe
1550 PROCvdu
1560 FORA%=0T06
1570   GCOL0,8+A%
1580   MOVE0,1023-143*A%
1590   DRAW1279,1023-143*A%
1600   PLOT85,0,880-143*A%
1610   PLOT85,1279,880-143*A%
1620   NEXT
1630 ENDPROC
1640 DEFPROCt
1650 GCOL0,1
1660 MOVE0,0
1670 DRAW0,1023
1680 MOVE640,0
1690 DRAW640,1023
1700 PLOT85,1279,0
1710 PLOT85,1279,1023
1720 GCOL0,0
1730 MOVE800,0
1740 DRAW800,1023
1750 ENDPROC
1760 DEFPROC2t
1770 VDU19,1,7,0;0
1780 VDU19,2,6,0;0
1790 PROCt
1800 GCOL0,2
1810 MOVE320,0
1820 DRAW320,1023
1830 PLOT85,336,0
1840 PLOT85,336,1023
1850 ENDPROC
1860 DEFPROCstar
1870 VDU20
1880 PROCs(640,511,512)
1890 PROCs(125,125,125)
1900 PROCs(125,898,125)
1910 PROCs(1154,125,125)
1920 PROCs(1154,898,125)
1930 VDU29,0;0;
1940 ENDPROC
1950 DEFPROCc(B%,C%,D%)
1960 VDU29,B%;C%;
1970 FORA%=0T0179STEP2
1980   MOVE0,0
1990   DRAW0,0
2000   PLOT85,D%*S(A%),D%*C(A%)
2010   PLOT85,D%*S(A%+1),D%*C(A%+
i)
2020   NEXT
2030 ENDPROC
2040 DEFPROCcircle
2050 VDU20
2060 PROCc(640,511,512)
2070 PROCc(125,125,125)
2080 PROCc(125,898,125)
2090 PROCc(1154,125,125)
2100 PROCc(1154,898,125)
2110 VDU29,0;0;
2120 ENDPROC
2130 DEFPROCc(B%,C%,D)
2140 VDU29,B%;C%;
2150 REPEAT
2160   E=D*0.93
2170   F% =5-E DIV175
2180   MOVE0,E
2190   DRAW0,D
2200   FORA%=0T0180STEPF%
2210     IFF%=1DRAWE*S(A%),E*C(A%
);GOTO2240
2220     PLOT85,E*S(A%),E*C(A%)
2230     PLOT85,D*S(A%),D*C(A%)
2240     NEXT
2250     D=E*0.88
2260     UNTILE-D<2
2270 ENDPROC
2280 DEFPROCstreak
2290 VDU20,19,1,7,0;0
2300 PROCbar
2310 FORA%=0T01
2320   GCOL0,1-A%
2330   FORB%=0T03
2340     C%=50+100*B%+512*A%
2350     MOVE500,C%
2360     DRAW500,C%+50
2370     PLOT85,500+40*2^B%,C%
2380     PLOT85,500+40*2^B%,C%+50
2390     NEXT: NEXT
2400 ENDPROC
2410 DEFPROCd(A%)
2420 MOVEX%+A%,128
2430 DRAWX%+A%,1023
2440 ENDPROC
2450 DEFFNb(B%,X%)
2460 C%=X%+127
2470 REPEAT
2480   FORA%=1T08%STEP2
2490     PROCd(A%)
2500     NEXT
2510     X%=X%+8%*2
2520     UNTILX%>C%
2530 =X%
2540 DEFPROCfreq
2550 X%=FNb(2,0)
2560 X%=X%+1
2570 REPEAT
2580   PROCd(0)
2590   MOVEX%+2,128
2600   PLOT21,X%+2,1023
2610   MOVEX%+6,132
2620   PLOT21,X%+6,1023
2630   PROCd(8)

```

Back in LASERBUG Issue 3 (June/July '82) on pages 20/21 we featured a popular screen dump PROCEDURE by Graham Anderson. As a follow up to this there are two programs below which perform faster (and much shorter in the latter case).

Following on from my earlier BASIC screen dump routine below is a machine-code version plus a program to test it and demonstrate how the routine can be called. It is assembled into the RS423 buffers but a change to line 280 will alter that. Once assembled it can be saved by \*SAVE "MCDUMP" or similar if assembled elsewhere. The lines of 1's, 2's and 3's were my own pitiful attempts to show the structure of the algorithm used. The algorithm is basically the same as that used in the BASIC version.

Graham Anderson

```

>
90 REM "ASSEMBLE AND TEST SCREEN
  DUMP ROUTINE
95 REM
100 MODE 0
110 PROCASSEMBLE
130 MODE 5
150 MOVE 0,0:DRAW 1279,0:DRAW 1279,1023:DRAW 0,1023:DRAW
  0,0:DRAW 1279,1023:MOVE 1279,0:DRAW 0,1023
154 MOVE 0,0
155 FOR I=1 TO 1250 STEP 50:DRAW I,1023:MOVE 0,0:NEXT I
160 colorZ=3:modeZ=5:CALL &0900,colorZ,modeZ
190 STOP
200 REM
210 REM *****
220 REM
230 DEF PROCASSEMBLE
240 oswrchZ=&FFEE
250 Parameters=&0600
260 LineZ=&70:VerticalspotZ=&71:XcoordZ=&75:YcoordZ=&77:
  BytevalueZ=&7A:BitvalueZ=&7B:colorZ=&7C:modfactorZ=
  &7D:TEMPZ=&7E:Bitcount
  Z=&7F
270 FOR optionZ=0 TO 3 STEP 3
280 PZ=&0900 : REM RS423 TRANSMIT AND RECEIVE BUFFERS
290 [OPT optionZ
294 \
295 \ GET PARAMETERS
296 \
300 LDA Parameters+1:STA TEMPZ
310 LDA Parameters+2:STA TEMPZ+1
330 LDX E0 : LDA (TEMPZ,X) : STA colorZ
340 LDA Parameters+4:STA TEMPZ
350 LDA Parameters+5:STA TEMPZ+1
360 LDA (TEMPZ,X)
364 \
365 \ CHECK MODE
366 \
370 BEQ mode0
380 CMP E1:BEQ modelor4
390 CMP E4:BEQ modelor4
400 CMP E2:BEQ mode2or5
410 CMP E5:BEQ mode2or5
414 \
415 \ MODE ERROR
416 \
420 BRK
421 ]
422 ?PZ=45 : REM Fault number (new one)

```

```

423 $(PZ+1)="Wrong mode" : REM Fault message
424 PZ=PZ+11
425 [ OPT optionZ
426 BRK \ Insert 00 as end of fault message indicator
427 \
428 \ SET MODEFACTOR
429 \
430 .mode0 LDA E1: STA modfactorZ:BNE prepareprinter
440 .modelor4 LDA E2: STA modfactorZ:BNE prepareprinter
450 .mode2or5 LDA E4: STA modfactorZ
460 \
470 \ PREPARE PRINTER
480 \
490 . prepareprinter
500 LDA E2:JSR oswrchZ \ enable printer
510 LDA E27:JSR printerchar \ adjust
520 LDA E65:JSR printerchar \ line
530 LDA E8:JSR printerchar \ spacing
550 \
570 LDA E31 : STA LineZ \ line counter

```

## formatting

Outputs from programs such as MORTGAGE (Issue 5, page 10) can be greatly improved through column formatting. Unfortunately, BBC BASIC provides no direct method of achieving this. There is a formatting parameter -@%, but this affects all output, including integers. To get around this I use the function FNformat:

continued on page 18

Electronics Applied, 4 Dromore Road,  
Carrick Fergus, Co. Antrim BT38 7PJ.  
Mail Order Only. Add 50p/order P&P

**SOFTWARE**

**BBC PADDLES WITH FREE ASTEROID  
BELT GAME — £11.50.**

**BBC**

*Envelope and Character  
Definer.* (32k)

Together these utility programs offer a complete character and sound envelope defining package for the BBC Micro. Even if you fully understand both the envelope and character defining commands, these programs will make their definition quicker and more accurate.

ALLOW 5 DAYS FOR DELIVERY

**ENVELOPE DEFINER**

• Excellent use made of graphics windows and colour to display patch and volume

• Database containing 20 predefined envelopes of everything from phasers to explosions

• Example graphs and step by step prompts allow easy defining of your own sound envelopes

Recorded on quality cassettes, sent by first class post

**CHARACTER DEFINER**

• A very nice VDU23 character definition program — Computer Users Club (GB).

• Shows both magnified and true size characters as they are defined

• All other definable characters can be called up for display

• Easy to use and check performed on all input

**INTRODUCTORY PRICE £4.95. Dealer Enquiries Welcome**

**(or just £3.35 + the incomplete cassette lead supplied with your Micro.)**

**CASSETTE LEADS**

a. 7 Pin Din to 2 x 3.5mm & 1 x 2.5mm Mini Jack

b. 7 Pin Din to 5 Pin Din & 2.5mm Mini Jack

c. 7 Pin Din to 5 Pin Din ..... £3.40

```

590 \
600 \ 1111111111 SCREENDUMP 111111111111111111111111111111111111>
610 \
620 .Nextline
630 LDA £27:JSR printerchar \ set up dual density
640 LDA £76:JSR printerchar \ bit image mode -
650 LDA £128:JSR printerchar \ 640
660 LDA £2:JSR printerchar \ bytes
700 LDA £0 : STA VerticalspotZ+1 \ Calculate
720 LDA LineZ : STA VerticalspotZ \ lowest
730 LDX £5 \ y-coordinate
770 .LAB2 ASL VerticalspotZ : ROL VerticalspotZ+1 \ for
780 DEX \ line
790 BNE LAB2 \ ( = LineZ*32)
800 \
830 LDA £0 : STA XcoordZ : STA XcoordZ+1
840 \ 22222222222222222222222222222222222222222222222222222222222222222222>
860 .Lineloop \ DUMP A LINE
920 LDA £8 : STA BitcountZ
930 LDA £0 : STA BytevalueZ
950 LDA £1 :STA BitvalueZ
1010 LDA VerticalspotZ+1 : STA YcoordZ+1
1020 LDA VerticalspotZ : STA YcoordZ
1030 \ 33333333333333333333333333333333333333333333333333333333333333333333>
1050 .Byteloop \ BUILD BYTE - SEND TO PRINTER
1090 LDY £00 : LDX £XcoordZ : LDA £9 : JSR &FFF1 \ perform POINT function
1160 LDA YcoordZ+2 : CMP colorZ : BNE LAB5 \ check colour
1190 LDA BytevalueZ : ORA BitvalueZ : STA BytevalueZ \ add bitvalue to bytevalue
1250 .LAB5 ASL BitvalueZ \ double BitvalueZ
1290 LDA YcoordZ : CLC : ADC £4 : STA YcoordZ : BCC LAB6 \ next y-coordinate
1340 INC YcoordZ+1
1380 .LAB6 DEC BitcountZ : BNE Byteloop \ end of byte?
1440 LDI modefactorZ \ send
1450 .sendbyte LDA BytevalueZ : JSR printerchar \ byte(s)
1460 DEX : BNE sendbyte \ to printer
1500 \ 33333333333333333333333333333333333333333333333333333333333333333333>
1510 LDX £2 : LDA XcoordZ \
1520 .LAB9 CLC : ADC modefactorZ : BCC LAB8 \ adjust

```

```

1530 INC XcoordZ+1 \
1540 .LAB8 : DEX : BNE LAB9 \ x-coordinate
1550 STA XcoordZ \
1566 LDA XcoordZ+1 : CMP £5 : BMI Lineloop \ check for end of line
1580 \ 22222222222222222222222222222222222222222222222222222222222222222222>
1690 LDA £13 : JSR printerchar \ print line
1730 DEC LineZ : BMI exit : JMP Nextline \ all lines done?
1750 \ 11111111111111111111111111111111111111111111111111111111111111111111>
1760 \
1770 \ TIDY UP BEFORE RETURNING TO BASIC
1780 \
1790 .exit
1800 LDA £27:JSR printerchar \ reset
1810 LDA £50:JSR printerchar \ line spacing
1850 LDA £3:JSR oswrchZ \ disable printer
1890 RTS \ exit to BASIC
1900 \
1910 \ ***** SUBROUTINE TO SEND CHARACTER TO PRINTER ONLY *****
1920 \
1930 .printerchar PHA : LDA £1 : JSR oswrchZ : PLA : JSR oswrchZ : RTS
1950 ]
1960 NEXT optionZ
1970 ENDPROC

```

## assembler prog – BBC micro pt 1

Having an assembler built into the BBC ROM makes machine code programming as easy as BASIC, but the manual is none too helpful on this. Assuming that you are already familiar with writing programs in BASIC you will already know the need for planning out things in advance – now all that is needed is the knowledge of how to put your program into machine code.

An assembler is a way of converting mnemonics that can be easily understood by humans, into the numbers that can be understood by the 6502 chip inside the BBC. It is important to understand that when you put some assembly code into your program the translation is all that occurs, to get your

# PRINT OUT



*“Life would be so much easier with an Amber 2400 Matrix Printer.....”*

**AMBER 2400  
MATRIX PRINTER**

LOW RUNNING COST USING  
PLAIN PAPER  
FITS BBC, DRAGON, SPECTRUM  
ATOM, TRS 80, UK101, VIC-20  
AND MOST OTHERS

---

**£69.95**

---

excluding VAT  
SEND LARGE STAMPED  
ADDRESSED ENVELOPE FOR  
DETAILS TO:

**AMBER CONTROLS LIMITED**  
Central Way  
Walworth Industrial Est.  
Andover  
Hampshire SP10 5AL

The other routine, from JEREMY RUSTON, present a one line alternative that works in 100 seconds!!! Two versions are given, one for MODE 4 and another for MODE 0. You should note that the one-liner's will not work on colour mode or when the screen has scrolled. It was written on OS 1.0 although should work on all versions.

```

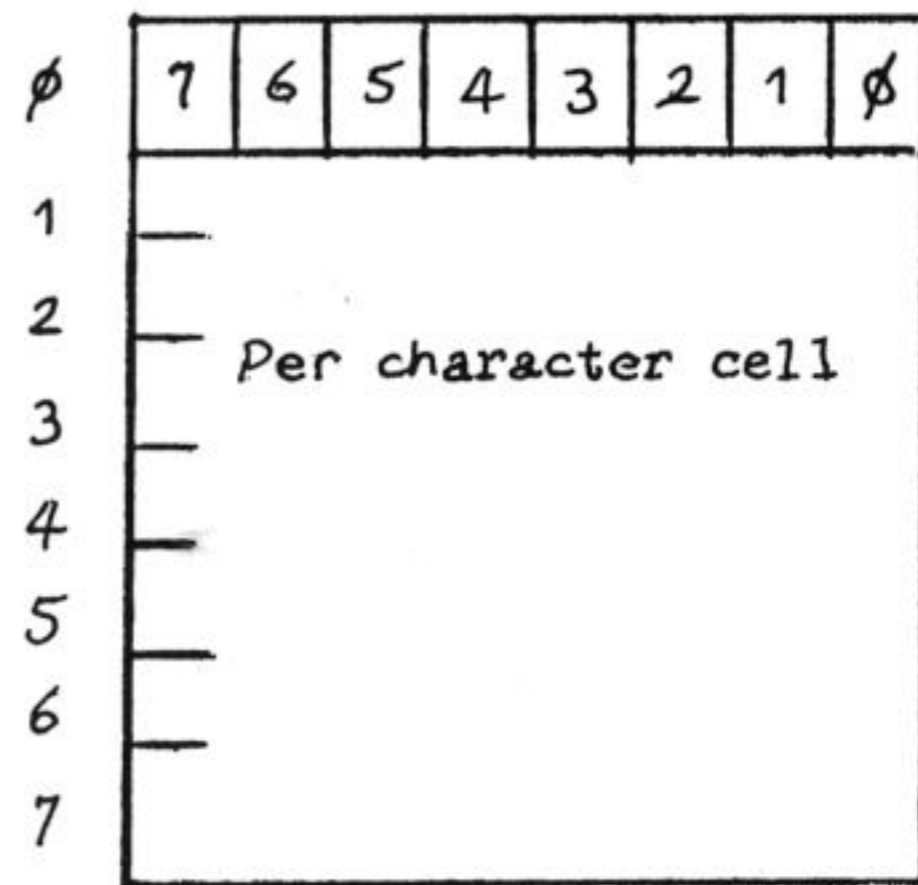
>LIST
10
20
30 REM One line screen dump program for the Epson MX80
40
50 REM Takes about 100 seconds
60
70 REM (c) 1982 Jeremy Ruston
80
90
100 REM Mode 4 -
110
120 VDU2,1,27,1,65,1,8:FORL%=0T039:VDU1,27,1,76,1,0,1,2:FORI%=31T00STEP-1:FORG
%:=7T00STEP-1:R%=?{HIMEM+T%*320+G%+L%*8}:VDU1,R%,1,R%:NEXTG%,T%:VDU1,13,1,10:NEXT
L%:VDU1,27,1,50,3:RETURN
130
140
150 REM Mode 0 -
160
170 VDU2,1,27,1,65,1,8:FORL%=0T079:VDU1,27,1,76,1,0,1,3:FORI%=31T00STEP-1:FORG
%:=7T00STEP-1:R%=?{HIMEM+T%*640+G%+L%*8}:VDU1,R%,1,R%:NEXTG%,T%:VDU1,13,1,10
:NEXTL%:VDU1,27,1,50,3:RETURN
180
190 VDU 24,640;0;1279;1023;
200
210 GCOL 4,129
220
230 CLG
240 GOSUB 170
    
```

continued from page 11

```

2640  X%=X%+12
2650  UNTILX%>254
2660  X%=FNb(4,X%)
2670  FORBX=6TO12STEP2
2680  X%=FNb(B%,X%)
2690  NEXT
2700  X%=FNb(16,X%)
2710  X%=FNb(32,X%)
2720  X%=FNb(32,X%)
2730  PRINTTAB(4,29);"8      5.3
      4      2.7      2      1.6
      1      0.5" SPC(38);"Mh
z,"
2740  ENDPROC
>
    
```

MODE 6 - 4 - φ



In MODEs 5 and 1 the situation starts becoming complicated. Out of the eight bits, pairs control the colours of four pixels at a time:

	Bits
1st Pixel	7 & 3
2nd Pixel	6 & 2
3rd Pixel	5 & 1
4th Pixel	4 & 0

What each pair of bits are set to determines what colour the pixel is:

Colour	1st Bit	2nd Bit
0 (Background)	0	0
1	0	1
2	1	0
3 (Foreground)	1	1

## rainbowspot

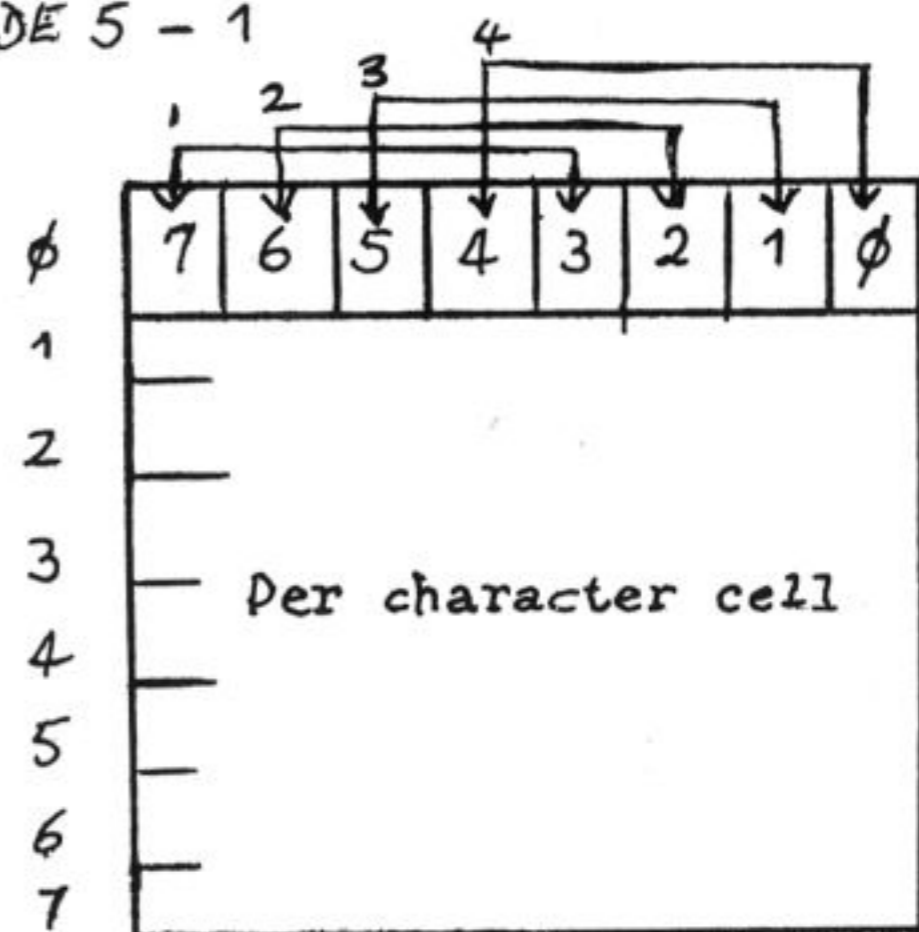
*Rainbowspot is the title of this occasional (for the moment) series. What is it about? Well, the colour capabilities of the BBC Micro is something we have explored in the past in various places – now it has an article all to itself. This month we take a look at how the colours are actually stored in the computer.*

The program on page 10 of LASERBUG Issue 3 was of great interest to me and below is my own, re-written version of this.

```

>L.
10  MODE2
20  *FX15,8
30  DIMA(15)
40  DIMB(16)
50  FORI=0TO14
60  READA
70  A(CI)=A
80  NEXT
90  FORI=0TO15
100 READB
110 B(CI)=B
120 NEXT
130 address=HIMEM
140 FORC=0TO15
150  c1=A(c)
160  FORD=0TO16
170  c2=B(d)
180  colour=(c1 AND170)+(c2 A
ND85)
190  address=address+160
200  FORPixel=address TO(addr
ess+160)
210  ?Pixel=colour
220  IFaddress>&8000THENadd
ress=HIMEM
230  NEXT
240  NEXT
250  NEXT
    
```

MODE 5 - 1



In MODE2 each byte controls two pixels with there being four bits per pixel. Each bit can be set in 16 combinations giving the 16 colours in MODE2:

Colour	1st Bit	2nd Bit	3rd Bit	4th Bit
0 (Background)	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1

etc. ....

For the first pixel the bits are 7, 5, 3 and 1 and for the second pixel they are 6, 4, 2 and 0.

Deciphering all that if you set one pixel to be RED and the second to BLUE a few times you get the colour purple. We would be interested in receiving a program for 3 and 4 colour maps.

**DICTIONARY TIME:** For beginners we will explain a few of the terms used in this article. A **PIXEL** is a single dot on the screen which can be one of many colours (depending exactly on what MODE you are in). Computers store all things in what is known as a **BYTE**. A **BYTE** is a number between 0 and 255 which is referred to in **BINARY**. **BINARY** is another number base in which all numbers are represented in 1's and 0's. A **BYTE** in **BINARY** is 8 digits long with a digit being known as a **BIT** (**B**inary **D**igi**T**). A **BYTE** looks something like this:

Both the original, and this one, work by virtue of the way colours are stored in the screen memory. Below are details of how the screen is stored in every MODE:

In MODE7 (the teletext MODE) there is one character code per byte and is simple enough to leave out for the purposes of this article.

In MODEs 0, 4 and 6 each binary bit represents 1 pixel on screen therefore each bit can be 0 or 1 making the pixel colour X or Y and hence two colours being available.

128	64	32	16	8	4	2	1

In each box can either be 1 or 0. If we wanted to write the number 159 in **BINARY** we would start off with the 128 box and work downwards. The number 159 is greater than 128 and hence you put a 1 (yes) in the first box. If you take away 128 from 159 you are left with 31. As 31 is less than 64 you put a 0 (no) in the 64 box and do the same with the 32 one. As 31 is greater than 16 you put a 1 (yes) in the 16 box which leaves 15. If we carry on this process we put a 1 in the 8 box, 1 in the 4 box, 1 in the 2 box and 1 in the 1 box. This means that 159 is equivalent to 10011111. For what we were doing in the article we need to work the other way around.

If we were working in **MODE 5** let's pretend we wanted to make the four **PIXELS** we can control to be colours 2, 1, 3 and 2 respectively. For colour 2 the **BITS** have to be 1/0, for colour 1 0/1 and for colour 3 1/1. Also the **BITS** we have to control are 7/3, 6/2, 5/1 and 4/0. For the first **PIXEL BIT 7** has to be set to 1 and **BIT 3** to 0, **BIT 6** set to 0 and **BIT 2** set to 1 and so on. The pattern we end up with is 10110110. This has to be converted into a normal number by reversing the process we carried out earlier which comes out to a grand total of 182 which must be **POKEd** into the correct location.

Michael Nott and Paul Barbour

*continued from page 13*

code to actually do anything you must **CALL** it. First, how to create the code from a **BASIC** program – try the following short program:

```
L.
 10 P%=&DF0
 20 [
 30 LDA#ASC("G")
 40 JMP&FFEE
 50 ]
```

When it is **RUN** you will see various letters and numbers appear on the screen, including what has been typed in at lines 30 and 40:

```
>RUN
0DF0
0DF0 A9 47 LDA#ASC("G")
0DF2 4C EE FF JMP&FFEE
```

Now to get the machine code to run try:

```
>REPEAT:CALL&DF0:UNTILFALSE
```

You should see the screen being covered with the letter G, until you press the escape key. This is because each time you execute the **CALL&DF0** you cause your machine code to be executed and this prints a G to the screen. To see why look back at your program.

Line 10 tells the machine to start the code at hexadecimal memory location 0DF0 (this is why **CALL&DF0** is used to execute it). Line 20 tells it what will follow are assembler mnemonics and line 50 tells it to stop the assembly process. When you write an assembler program it should always be preceded by a square open bracket (shown as a left arrow in **MODE 7**) and ended by a close square bracket (shown as a right arrow in **MODE 7**) but what goes in between is up to you, providing that they are valid 6502 mnemonics. The best way to learn these is probably from a book such as 6502 Assembly Language Programming by Lance Lerenthal published by McGraw Hill but failing that I hope that my articles in **LASERBUG** will prove extremely useful (Personal Software, Winter '82 has a 21 page article on 6502 assembler. Even though it was really written to help people with **PET** assembler I feel the article would be useful to all who are trying to get to grips with assembler. If nothing else it lists all the 6502 mnemonics with explanations of them. It is available from your local newsagent – Ed.)

It maybe useful to think of your programs in terms of **BASIC** until you get the hang of things, converting short sections into assembler. You may be sick of people telling you that your programs should be structured but if it is important in **BASIC** it is essential in machine code. Always write short subroutines and if you need a long piece of code do it by linking these routines together. In this way it should be possible to check each routine separately so that the inevitable bugs can be tracked down.

A subroutine will take the form of:

```
L.
 500 [NAME%
 610 RTS:]
```

Note the use of a colon to separate the end bracket from the return from subroutine mnemonic; it can also separate instructions on a single line as in **BASIC**. **NAME%** is a variable. When the program reaches the line it gives to that variable a value which points to the memory where the following code will be placed. An equivalent way of doing this is:

```
>L.
 500 DIM NAME% 100
 510 P%=NAME%
 520 [.....
 530 RTS:]
```

This is more useful since the first line reserves 101 bytes (from 0 to 100) and points to the first byte with **NAME%**. The second line ensures that the space reserved will be used to store the machine code created by the assembler.

Now a quick example to end with. Suppose we have a **PROCEDURE** to printout the next character input from the keyboard. We may do this by:

```
>L.
 10 DEFPROCprntchr
 20 LOCAL A$
 30 A$=INKEY$(0)
 40 PRINTA$)
 50 ENDPROC
```

or perhaps you would use:

```
>L.
 10 DEFPROCprntchr
 20 LOCAL A%
 30 A%=INKEY(0)
 40 VDUAR%
 50 ENDPROC
```

Both would have the same effect, but it is the latter that we shall convert to assembler. We can use the operating system subroutines for input and output as outlined in the user guide.

```
>L.
 10 OSWRCH=&FFEE
 20 OSRDCH=&FFE0
 30 DIM CODE 20
 40 P%=CODE
 50 [JSR OSRDCH
 60 JSR OSWRCH
 70 RTS:]
```

**CALL CODE** will then have the same effect as **PROCprntchr**. To understand why, it is necessary to know that **JSR** means jump to a subroutine; when an **RTS** instruction is found execution continues from the next instruction after the **JSR**. The **BASIC** equivalents are **GOSUB** and **RETURN**. **OSRDCH** and **OSWRCH** are in the operating system ROMs and are documented in the User Guide. To give some idea of what they do, think of the "registers" (known as A, X and Y) as variables. Most of machine code programming concerns using these registers to cause the required effect. I will be explaining how next month.

Nick Goodwin



This month we review the Amber 2400 printer which eventually found its way onto my desk. It is manufactured by the Hampshire based company Amber Controls Ltd, and costs £69.95 + VAT + P&P + a suitable cable. If you are using the parallel port (which I assume you would) this will cost you a grand total of £96.80. The printer is only 24 columns but uses calculator paper which is readily available.

If you have a Model A please do not stop reading, - I will explain how you can get the Amber working on your computer later.

As we have explained before VDU2 or CTRL-B turns on the printer and VDU3 or CTRL-C turns it off again. Assuming you are using the parallel interface, no other commands are necessary to get your printer working.

On the point of turning the printer on and off (in real terms) there is no power switch on the Amber and hence you have to resort to the drastic action of pulling out the plug.

The entire character set of the printer is listed for you below:

32 -	33 - !	80 - P	81 - Q
34 - "	35 - #	82 - R	83 - S
36 - \$	37 - %	84 - T	85 - U
38 - &	39 - '	86 - V	87 - W
40 - (	41 - )	88 - X	89 - Y
42 - *	43 - +	90 - Z	91 - [
44 - ,	45 - -	92 - \	93 - ]
46 - .	47 - /	94 - ↑	95 - _
48 - 0	49 - 1	96 - `	97 - a
50 - 2	51 - 3	98 - b	99 - c
52 - 4	53 - 5	100 - d	101 - e
54 - 6	55 - 7	102 - f	103 - g
56 - 8	57 - 9	104 - h	105 - i
58 - :	59 - ;	106 - j	107 - k
60 - <	61 - =	108 - l	109 - m
62 - >	63 - ?	110 - n	111 - o
64 - @	65 - A	112 - p	113 - a
66 - B	67 - C	114 - r	115 - s
68 - D	69 - E	116 - t	117 - u
70 - F	71 - G	118 - v	119 - w
72 - H	73 - I	120 - x	121 - y
74 - J	75 - K	122 - z	123 - <
76 - L	77 - M	124 - !	125 - >
78 - N	79 - O	126 - ~	127 -

The most noticeable omission from the character set is the lack of a £ sign. However for £100 you can't expect everything!

The facility for different sizes of text is included in this printer which is controlled by 3 different control codes. As well as normal text double height text, double width text and double width and height text are possible. Examples of these are shown:

THIS IS NORMAL TEXT

THIS IS  
DOUBLE WIDTH  
TEXT

THIS IS DOUBLE HEIGHT  
TEXT

THIS IS  
DOUBLE WIDTH  
AND HEIGHT  
TEXT

Normal size text is obviously the default option and uses a 5 (w) x 7 (h) matrix. Double width text with a 10 x 7 matrix is obtained by sending 14 or &E to the printer (using VDU1,14). Double height text has a 5 x 14 matrix and is operated by sending 18 or &12 to the printer. Both double height and double width text is obtained by sending BOTH codes to the printer i.e. VDU1,14,1,18. Normal text can be obtained again by VDU1,20 (or VDU1,&14) which turns off all the control codes.

Before I carry on with the review, I must just explain how the Amber printer works as it is very unusual. The printer prints one horizontal line at a time, 5 (or whatever) times to form the characters. A line of text is made up

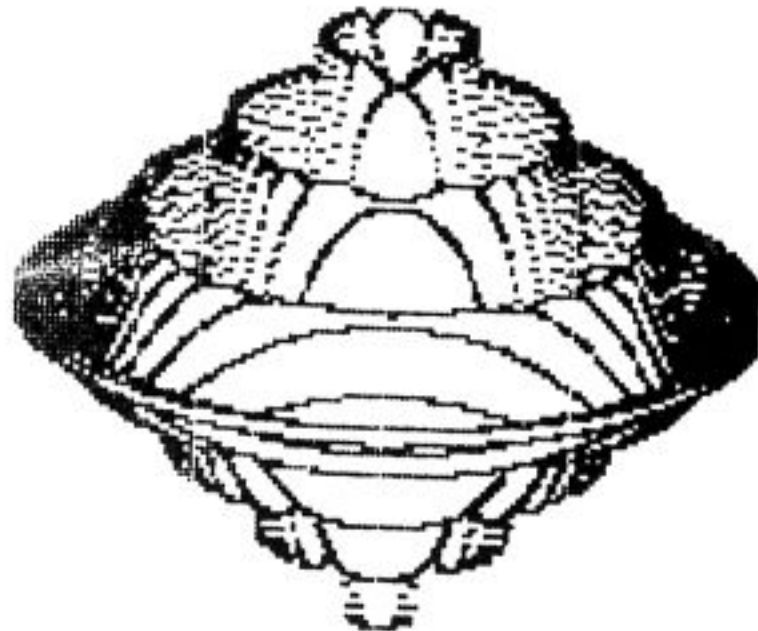
of 24 matrices with each matrix being 5 x 7. This means that the printer has a matrix 120 x 7 in its memory. Most printers would print the first 5 x 7 matrix, then the next and so on. However Amber prints the entire top row of 120 pixels, then the next, until the full 7 lines have been printed.

Understanding this method is essential if you are to get to grips with the graphic capabilities of the machine. To enter graphics into the printer you have to enter them a single line at a time (i.e. all 144 pixels) The extra 24 pixels come in in graphics mode because in text, these 24 provided the spaces between the words. Each line must have the code 17 or &11 preceding it to indicate that the line will contain graphics, not text. The next 18 characters sent to the printer must represent 8 pixels (i.e. an 8 bit word) with the pixels being in the order 128/64/32/16/8/4/2/1. If all this seems a bit complicated to you don't worry - LASERBUG will come to the rescue. If you are using either MODE 2 or 5 on the computer, below is a screen dump procedure especially for the Amber.

```

LIST
9890 REM Screen dump
9900 REM PROC for Amber
9901 REM 2400
9910
9920 REM (C) LASERBUG
9930 REM 1982
9940
9950 REM Works on MODEs
9960 REM 2 & 5 only
9970
9980 .....
9990
10000 DEFPROCscreen_dump
10010 VDU2:PRINT
10020 FOR Y%=1023 TO -8 STEP
-8
10030 VDU1,17
10040 FOR X%=0 TO 1280 STEP 7
1.1111111
10050 echar%=0
10060 RESTORE 10170
10070 FOR P%=X% TO (X%+62.2
222222) STEP 8.88888889
10080 READ bin%
10090 add%=FALSE
10100 FOR block%=P% TOP%+8
10110 IF POINT(block%,Y%)
>0 THEN add%=TRUE
10120 IF POINT(block%,Y%+
4)>0 THEN add%=TRUE
10130 NEXT
10140 IF add%=TRUE THEN e
char%=echar%+bin%
10150 NEXT:VDU1,echar%
10160 NEXT:NEXT
10170 DATA 128,64,32,16,8
,4,2,1
10180 PRINT:VDU3
    
```

The PROCedure is very effective as can be seen from the sample screen dump:



**HIMEM:** Returns the highest memory location available before the graphics.  
**RAM:** Unique to BBC Micro and would have to be found out on other micros using PEEKs and POKEs.

**IF:** Standard BASIC.

**INKEYS:** The exact use varies between computers. On some INKEYS replaces GETS. On others it has a variable which determines the length of time to wait for input. On the BBC Micro INKEY can have a negative argument which is the same as using PEEKs to read the keyboard on other micros.

**INPUT:** Standard BASIC.

**INPUT##:** Standard file handling command inputting a string from file. Available in some form on most micros.

**INSTR:** Searches for one string inside another. Unique to BBC Micro and would have to be replaced by a specially written subroutine on other computers.

**INT:** Returns the integer value of its argument. Standard BASIC.

**LEFT\$:** Standard BASIC although on some micros takes another form e.g. string (TO n) on the ZX81/Spectrum.

**LEN:** Standard BASIC.

**LET:** Standard BASIC although optional on most machines.

**LLIST:** Lists specified lines out to printer. Not available directly on BBC Micro (VDU2 (RETURN) LISTm,n (RETURN) VDU3 (RETURN)).

**LISTO:** Sets list options. Unique to BBC Micros.

**LN:** Returns natural logarithm on its argument. Normally called LOG on most other micros.

**LOAD:** Standard BASIC. Called CLOAD on some computers.

**LOCAL:** Defines a variable to be local to a procedure. Unique to BBC.

**LOG:** Returns common logarithm of its argument. Not normally implemented on other micros.

**LOMEM:** Returns the first memory location available after variable storage. Unique to BBC. Would have to be found out using PEEKs and POKEs on other computers.

**MID\$:** Fairly standard BASIC but not available on all micros. Sometimes referred to by another name i.e. string (m TO n) on Spectrum.

**MOD:** Gives the modulus of its argument. Not normally implemented.

**MODE:** Sets what screen MODE the computer is to operate in. Unique to BBC.

**MOVE:** Moves the graphics cursor to position x,y. Unique to BBC.

**NEXT MONTH:** N to S.

Paul Barbour

## corrections

### ISSUE 5 (October)

In the Business Special article it was not stated that (i) the program was primarily designed for use for the Seikosha GP-100A (although it should work well on any printer) and (ii) that you had to press the COPY key twice to copy out the screen onto the printer. Also for the program to work properly you must add another line:

```
1135 A%=135
```

On pages 17/18 it was not stated that the joystick article assumes you are using them with the fire button towards you. Although technically this is the wrong way round it is accepted that this is the way most people use them – even the Acornsoft games that use joysticks assume this.

In the article on RGB Conversions (pages 28/29) a series of X's seems to have crept in from somewhere???

On the membership form (page 31) a small typing error slipped our notice. Please note that Colnbrook is in Slough, not Slouth.

### ISSUE 6 (November)

First of the corrections in the November issue refers to the Editorial. The price of the Torch Computer was not "around £2,000" but £4,019. However it has just been announced that Torch have lowered the price down to £2,795. Also the disk pack is lowered in price from £1,144 to £895. Further down that article was a totally incorrect paragraph about our dust covers. All these prices were out of date and were increased due to additional costs not given in the original quote we had. Sorry if we confused anybody.

On the software review of Alphabeta by H & H Software on page 13 we gave the program only two stars for value for money. This is incorrect and should have been three stars. My apologies to H & H Software.

On the following page was Phil Hirst's article on software protection. Towards the end it told you to type in P.N?&2F5. The N was incorrect and should have been the "squiggle" to the left of the double bar symbol on the keyboard.

The Wordprocessor Update program on page 15 had one line missing. You need to add:

```
755 IFC=142THENPROCback__space:GOTO486
```

We want to encourage the setting up of local groups under the general banner of LASERBUG. All it needs is for willing people to allow us to print their names and addresses (or telephone numbers) along with when their meetings take place to allow other people in turn to contact them with a view to either attending meetings or helping to organise them. It could be as small as a couple of people in someone's front room or a large affair at the local church hall or school.

Generally we want to leave this up to you. Of course we are willing to give advice and help you where we can but the main idea is to get local groups functioning by themselves. We would like to see a whole network of local groups holding regular meetings so that anyone in possession of a BBC MICRO can get together with other owners/users wherever he lives.

We at LASERBUG feel very strongly that belonging to a local user group is the best way to get the most out of your computer. As well as meeting new friends you can get the benefit of other peoples discoveries – and share your own with them. If there is not one near you then why not start one yourself?

If you have started or are planning to start your own group then please let us know. Why not (assuming you have attended one of these meetings) write a short article on your experiences. We would be pleased to publish it on a Club Report page.

The present list of local user groups is:

John Claydon,	North London BBC Microcomputers Users Group and Education Workshop, Bounds Green Junior School, Park Road, N11. 01-889 5446. Meetings held on the second Sunday of each month at 2.00. Fee of approx. £1/meeting to cover costs.
G.W. Goodacre,	CHELMERBUG, 34 Quilp Drive, Chelmsford, CM1 4YA. Formal meetings held on the first Wednesday of each month at a local school. Informal meetings in-between in members homes.
Nick Lamb,	23 Gaywood Close, Caister-on-Sea, Great Yarmouth, Norfolk, NR30 5RD. 0493-728442.
Norman Lambert,	Orpington Computer Club, 11 Vinson Close, Orpington, Kent, BR6 0EQ. Meetings held every Friday evening at a local church hall.
Janne Soderberg,	Frihetsvagen 32, S-175 33 Jarfalla, Sweden. Phone 0758-317 53. The Scandinavian user group caters for both Atom and BBC owners.
Dr. P. Voke,	39 Ligo Avenue, Stoke Manderville, Aylesbury, Bucks, HP22 5TY. 0296-613974.

To be included in this list please write to us at our usual address giving full details and mark the envelope MEETINGS. A free affiliation scheme is available to all user groups and those interested should get in touch.

continued from page 12

```
L*
10000 DEFFNformat(r,w%,d%)
10010 REM r%=value to be formatted
10020 REM w%=width of field
10030 REM d%=number of decimal
```

continued on page 21

## errors?

Ever had an error on your BBC Micro? Of course you have. Well, now type in the following program from John Murphy of Telford. No clues given . . . .

```
L*
10 !&D26=&60606142;!&D2A=&70752
073
20 ?&2DE=0:PX=&D00:LOPT0
30 LDA#&B:STA&202:LDA#&D:STA&20
S
40 RTS:PHA:LDA#&97:CMP#FE:BNE&2
D0
50 LDA#&F1:CMP#FD:BNE&D00:LDA#&
25
60 STA#FD:LDA#&D:STA#FE:PL9
70 JMP#B&39:LDA#4:3
80 CALL&D00
90 x yes 'x'
```

continued from page 17

The Amber printer is fairly small and compact (although large when compared to the Sinclair printer) and although doesn't match the computer's colour, does fit in with the official cassette recorder. As described earlier the paper it uses is readily available and is the same as most printout calculators take. The ribbon must be bought from Amber themselves and costs £1.90. Whilst doing the review I didn't quite finish off one roll of paper but the printout had become rather faded – a bit too faded too soon for my liking, however the print was still easily readable. If you are using the printer for an considerable length of time or almost continuously the solenoids become over hot and produces 'waves' across the paper (a few lines might be indented) – the effect I am sure you have seen on the Sinclair printer in one of the listings published using it, however the problem on the Amber is nowhere near as bad.

What if you have a Model A? Well, Amber Controls offers upgrade kits for your BBC Micro so that you can have either the parallel or serial port (or both!) The upgrade kit cost £11.44 for the parallel version and £14.37½ for the serial. Both upgrade kits consist of two chips and the appropriate connector with of course instructions. The fitting instructions are very good and all that is required (except for a steady hand!) is "a good clean soldering iron with a properly earthed bit". You do need some experience with a soldering iron to be able to upgrade your computer safely without damaging it.

The Amber 2400 is a very good printer for the money and is suitable as the main printer for the smaller micros (i.e. ZX81, etc.). However the big question must be with a computer as big as the BBC Micro which is easily suited for full size printers, what is the Amber's use? Well, the first and most obvious answer to this is that people who want a printer for their BBC Micro but cannot (or don't want to afford) £250 at least would be interested in the Amber. This aside who else would be interested in it?

I personally feel a good number of people would be, mainly in professional situations. At times having an 80-column printout is a pain when 24 columns would do amply. For example, if you are in a school and wish to keep a hard copy of each child's results in a maths test say, the Amber would be fine:

## MATHS TEST 1

RESULTS OF:  
MARK BARBOUR

SCORE:  
28/38

PERCENTAGE:  
74

GRADE:  
B

COMMENTS:  
Good result  
Needs practice on  
8x tables

If you were doing scientific tests, the Amber would probably be suitable for printing up your results – even complex graphs using the graphics capabilities of the Amber. If you were doing stock control or something similar, the Amber again would be fine here.

Let me summarise. The Amber printer is very good value for money, uses readily available and cheap consumables and is fairly reliable. Its use is two-fold. Firstly as the main printer for somebody who cannot afford at least £250 for a full size one, i.e. a hobbyist who is still recovering from the £400 bill for the micro itself or say a school that is trying to kit out a whole classroom. The other use is for people who do not require an 80 column printout, be it because it is too big or whatever, and a 24 column would suffice.

The Amber is a good printer and I would recommend it to anyone who fits into the above categories.

PAUL BARBOUR

**MANUFACTURER:** Amber Controls Ltd., Central Way, Andover, Hants, SP10 5AL.

**PRICE:** £98.60 (including parallel cable, VAT and P&P)

**CHARS/LINE:** 24 (12 in double width mode)

**CHARS/SEC:** approx 17 chars/sec

**CHAR. MATRIX:** 5 x 7

**INTERFACE:** Serial & Parallel

**CHARACTER SET:** 94

**TRUE DESCENDERS:** No

**GRAPHICS RESOLUTION:** 144 x n lines

**PAPER FEED:** n/a

**COPIES:** n/a

**PAPER WIDTH:** 2¼" (normal calculator paper)

**SIZE:** 80 x 160 x 160mm

**WEIGHT:** ???

We would like to thank Amber Controls for loaning us the Amber 2400 for this review.

## focus on adventures

*As promised last month we are going to take a look at adventure programs in general, explain what they are about and the different types available. We will not go into too much depth about any particular one but at the end of the article is a list of some of the programs available which we will be reviewing in future months. Other ones that we will review depends entirely on the producers of the software.*

What is an adventure? The dictionary describes adventure as "a risky undertaking of unknown outcome/an exciting or unexpected event or course of events". All these things are true to the computer adventurer. Basically in an adventure program you are taken to a mystical place with a specific task to do. Whilst attempting to perform these tasks you are subject to all sorts of peril and danger.

Adventures are on the whole completely text based, the reason for which dates back to their computer origin. Adventure programs were originally started on large mainframe computers which used a teletype as the communications peripheral. Hence the program could only be text based. For the standard type of adventure a graphical display is of little help and so if there is any, this normally is just a map.

Adventures are one of the original computer games and can arguably be called the most popular game written. One version of Adventure runs in 120k on a mainframe with disks! You can be assured however that versions for your BBC Computer are very faithful to the original and some even have more rooms, etc. than the 120k mainframe version!

So who would play Adventure and how? An Adventure is basically a long puzzle. It is extremely unlikely that you could load in the program and solve it within an hour. An adventure is the kind of program/puzzle where you will play it for an hour and then come upon a seemingly impossible to solve problem. You will then put the program away and a few days later think of the answer, try again and get a little further. If you are the kind that enjoys an occasional flick through a puzzle book then you will probably enjoy an Adventure.

How do you play it? Generally the computer gives you a few lines of text telling you where you are, what is with you and any other information you need. For your actions you have to enter a sentence in English to describe what you want to do e.g. MOVE LEFT or TAKE ARROW or PULL LEVER.

The scenario in each Adventure is different. For example in FIRIENWOOD by MP SOFTWARE "an evil wizard had captured the magic Golden Bird Of Paradise and has imprisoned it in a weird castle in the middle of the enchanted Firienwood. Your quest is to find the bird and set it free, in return for its freedom the bird will give you health, wealth and prosperity. You will need magic to overcome the wizard's powers and before you can enter Firienwood safely, you will need to find a magic lamp. Beware! Many perils lie before you and every move is fraught with danger!!" This is typical of the task you have to complete.

As the Adventure generally will take you more than one sitting you have the facility in most programs to SAVE the game where you are and LOAD it back in later. This is essential and you should make sure that any Adventure you buy has this feature.

Most of the Adventures available are of the standard type as described above. A list of some of the ones available appears below:

COLOSSAL ADVENTURE – Level 9 Computing, 229 Hughenden Road, High Wycombe, Bucks.

ARROW OF DEATH – Digital Fantasia, 24 Norbreck Road, Blackpool.

FIRIENWOOD – MP Software & Services, 165 Spital Road, Bromborough, Merseyside, L62 2AE.

THE GOLDEN BATON – Digital Fantasia.

PHILOSOPHER'S QUEST – Acornsoft, 4a Market Hill, Cambridge, CB2 3NJ.

TIME MACHINE – Digital Fantasia.

All of the above programs are text based and work in MODE7. Some use colour in the teletext mode but the better ones because of their length do not even use this.

continued on page 23

This month's seasonal program combines a graphics display with a simple game. You must guess what picture the computer has drawn. The identity of the picture has been hidden as much as possible in the program, however it wouldn't require too much intelligence to cheat. Good luck and as a hint, think of last winter.....

Paul Barbour

LIST

```

10 REM      WINTER SCENE ?
20 REM (Seasonal 4 - January)
30 REM      by Paul Barbour
40 :
50 REM      17/12/82
60 :
70 REM      Version 1.0
80 :
90 REM Takes up ~2.45k memory
100 REM (~6.06k with variables)
110 REM and uses MODEs 5 and 7
120 REM (MODEs 5 and 7 for 16k)
130 :
140 REM      Suitable for 16k if
150 REM      all REM's are omitted
160 :
170 REM      Written on OS 0.1
180 :
190 REM      (C) LASERBUG 1983
200 :
210 : : : : :
220 :
230 #KEY10OLD:MOCLS:IM
240 ONERRORMODE7:END
250 MODE7:VDU23;8202;0;0;0;
260 PROCintro
270 MODE5:VDU23;8202;0;0;0;
280 PROChide
290 PROCshape
300 PROCcheck
310 GOTO270
320 END
330 :
340 : : : : :
350 :
360 DEFPROCintro
370 PRINTTAB(12,10);CHR#141"WINT
ER SCENE ?";TAB(12,11);CHR#141"WIN
TER SCENE ?"
380 PRINTTAB(12,14);CHR#141"by P
aul Barbour";TAB(12,15);CHR#141"by
Paul Barbour"
390 A=INKEY(500)
400 CLS
410 PRINT" ";CHR#157'" ";CHR#157
;TAB(11);CHR#132;CHR#141;"WINTER S
CENE ?"'";CHR#157;TAB(11);CHR#13
2;CHR#141;"WINTER SCENE ?"'";CHR
#157
420 PRINT'CHR#134;"      This pr
ogram turns a sketch into"CHR#134
;"a game.  When you start the comp
uter"
430 PRINTCHR#134;"will draw a Pi

```

```

cture without you seeing.");CHR#134
;"Then dots will appear randomly a
ll over";

```

```

440 PRINTCHR#134;"the screen EXC
EPT where the picture was";CHR#134
;"drawn.  When you think you know
what"

```

```

450 PRINTCHR#134;"the picture is
press the space bar."

```

```

460 PRINT'CHR#130;"Press RETUR
N when you are ready..."

```

```

470 REPEATUNTILINKEY(-74)

```

```

480 CLS

```

```

490 PRINTCHR#134;"The picture wi
ll take approximately 15"CHR#134"
seconds to draw, then you can try
to"

```

```

500 PRINTCHR#134;"guess what it
is."

```

```

510 PRINT'CHR#134;"Press the spa
ce bar WHEN you know what"CHR#134
;"the picture is."

```

```

520 TIME=0:REPEATUNTILTIME=800

```

```

530 ENDPROC

```

```

540 :

```

```

550 DEFPROChide

```

```

560 VDU19,3,0,0,0,0

```

```

570 MX=&2000:NY=&8196

```

```

580 !MX=&574F4E53

```

```

590 !NY=&D4E414D

```

```

600 ENDPROC

```

```

610 :

```

```

620 DEFPROCshape

```

```

630 PROCshape1(640,256,256)

```

```

640 PROCshape1(640,512,128)

```

```

650 GCOLOR,0

```

```

660 PROCshape1(600,542,10)

```

```

670 PROCshape1(680,542,10)

```

```

680 PROCshape1(640,522,10)

```

```

690 PROCshape1(640,100,10)

```

```

700 PROCshape1(640,200,10)

```

```

710 PROCshape1(640,300,10)

```

```

720 PROCshape1(640,400,10)

```

```

730 GCOLOR,3

```

```

740 PROCshape2

```

```

750 PROCshape3

```

```

760 GCOLOR,2

```

```

770 REPEATPROCdot

```

```

780 UNTILINKEY(-99)

```

```

790 ENDPROC

```

```

800 :

```

```

810 DEFPROCshape1(X%,Y%,Z%)

```

```

820 VDU29,X%;Y%;

```

```

830 FORA%=&0TO360STEP30

```

```

840 MOVE0,0

```

```

850 DRAW$INRAD(A%)*Z%,COSRAD(A
%)*Z%

```

```

860 PLOT65,SINRAD(A%+30)*Z%,CO
SRAD(A%+30)*Z%

```

```

870 PLOT85,0,0

```

```

880 NEXT
890 ENDPROC
900 :
910 DEFPROCshape2
920 VDU29,0;0;
930 MOVE500,600:DRAW780,600
940 MOVE530,600:DRAW530,650
950 PLOT85,750,650:DRAW750,600
960 PLOT85,530,600
970 ENDPROC
980 :
990 DEFPROCshape3
1000 MOVE840,300:DRAW840,750
1010 FORX%=-100TO100STEP25
1020 MOVE840,750:DRAW840+X%,950
1030 NEXT
1040 :
1050 DEFPROCdot
1060 REPEATX%=RND(1280):Y%=RND(10
23)
1070 UNTILPOINT(X%,Y%)=0
1080 PLOT69,X%,Y%
1090 ENDPROC
1100 :
1110 DEFPROCcheck
1120 CLS
1130 VDU20
1140 PRINT"WHAT WAS THE SHAPE"
1150 #FX15,0
1160 INPUT' 'shape$
1170
1180 REM Needed because of a bug
1190 REM in OS 0.1 - see User
1200 REM Guide, Page 281
1210 IFLEN(shape$)<LEN(#M%)THEN12
40
1220
1230 IFINSTR(shape$,#M%)>0THEN128
0
1240 CLS
1250 PRINTTAB(6,12):"WRONG !"
1260 TIME=0:REPEATUNTILTIME=300
1270 ENDPROC
1280 CLS
1290 PROCshape
1300 REPEATUNTIL0
>

```

continued from page 18

```

10040 REM Places (@ for integer)
10050 r%=INT(r#10^d%+.5)
10060 r$=STR$(r%)
10070 len%=LEN(r$)
10080 IFc%=0THENp%=0ELSEp%=1
10090 sp%=w%-len%-p%
10100 IFsp%>0THENsp$=STRING$(sp%, "
" )ELSEsp$=""
10110 IFd%=0THEN=sp#+r$ELSE=sp#+LE
FT$(r$,len%-d%)+", "+RIGHT$(r%,d%)

```

This converts all reals into integers - hence it is subject to the maximum value that the machine can handle. The function will handle, however, up to eight significant figures. For financial programs this is adequate - unless you

continued on page 23

## EPSON TYPE 3 PRINTERS

SAVE £60-£80 (+ VAT) when you order your superb new Type 3 printer from Datatech, the Epson specialists.

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

For FREE BROCHURE and special DISCOUNT OFFER write now to:

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

## competition 3 results

The mail for our third competition was the best we have had so far and I would like to thank all of you that entered.

There are several ways of getting the listing shown but we wanted one way in particular. Firstly though some of the other methods. A good number of people thought that we had just added the appropriate number of spaces to make the extra line 0 - WRONG. A smaller percentage of the wrong entries thought that we had merged two programs without renumbering using the TOP-2 method given in the user guide - WRONG. One person (Dr. Susans, a regular contributor to LASERBUG) thought that we had POKED in the extra line - WRONG AGAIN.

For the answer that we wanted we left two clues - (i) the way that the lines were numbered (32763-32767) which were the last possible lines in the program and (ii) the fact that they were using OS EPROM 0.1. Another clue could have been this magazine's title, laserBUG.

In OS EPROM 0.1 there is a bug (in fact there are many) which effects the RENUMBER command. For how you make the listing I will give you the winners explanation.

The answer is to type in the first line numbering it 10 and continuing:

```

10REM THIS IS THE PROGRAM
20REM THIS IS THE PROGRAM
30REM THIS IS THE PROGRAM
40
50
60REM THIS IS THE COPYRIGHT

```

and then to enter:  
RENUMBER 32763,1

as a direct command and LIST the program and the listing appears as required (lines 40 and 50 are obtained by pressing the space bar once having typed in the line number).

Well done to Philip Brown of Northwood, Middlesex who now has his membership extended to issue 17. This method is interesting in the fact that the listing has two line zeros of which only the real first one can be listed and altered. The false line zero at the end cannot be easily altered without RENUMBERING the program.

Once again thanks to all those who attempted the competition and better luck next time to all who didn't get it right.

Back in August I acquired the 1.00 Operating System and Disc Interface. The 1.00 OS is in EPROM and consists of a couple of 2764 EPROMS mounted on a small PCB. This has a configuration of pins beneath it to enable it to be plugged into the existing OS 'hole', (IC51). As soon as it is plugged in the facilities of 1.00 are available. The Floppy Disc Controller chip plugs into IC78 and the DFS ROM (Actually it is in EPROM) plugs into IC88. There are also another nine IC's which are included in the Interface Kit and have to be fitted in the area around the Disc Drive socket (PL8).

When the machine is powered up the following appears:

BBC Computer 32K

Acorn DFS

BASIC

>-

The DFS obviously stands for Disc Filing System. There are now a whole new series of commands available, and I shall briefly explain each one. One thing that is important is that once the Interface is fitted everything defaults to a Disc Drive. Therefore if you are working with the Cassette Filing System every time you power up or do a Hard Reset (which in 1.00 is pressing CTRL BREAK) then you must type in \*TAPE or \*T. for short. Incidentally with the "\*" commands the suffix can be in lower case eg. \*tape and it will still work! (we know, see LASERBUG Issue 4, page 6 - Ed.)

One very important thing which I feel is very wrong is that you can only get a copy of the Disc System User Guide if you buy a BBC Drive. I think the Guide should come with the DFS Interface as it is this DFS which the commands apply to. One does not have to use the BBC Drive to be able to use all the facilities. In fact I have been using a Teac Drive and have found no problems.

Anyway here are the DFS commands. I have not included the syntax as anyone who gets a DFS will be able to discover the syntax by using \*HELP.

*ACCESS	*A.	Locks or unlocks a file.
*BACKUP	*BAC.	Copies all information from one disc to another.
*BUILD	*BU.	Causes all subsequent keyboard entries to be stored in the named file.
*CAT n	*	Displays a disc catalogue. (n is the Drive number).
*COMPACT	*COM.	Collects all files on a disc together into one continuous sequence leaving a single block of free space.
*COPY	*COP.	Copies all specified files from one disc to another.
*DELETE	*DE.	Removes a single named file from a disc.
*DESTROY	*DES.	Removes specified files from a disc in a single action.
*DIR	*DI.	Changes the current set directory.
*DISC	*D.	Selects the Disc Filing System.
*DRIVE	*DR.	Changes the current set Drive.
*DUMP	*DU.	Produces a hexadecimal listing of a file.
*ENABLE	*EN.	Allows the use of *BACKUP and *DESTROY.
*EXEC	*E.	Reads a disc file byte by byte as if the bytes were being typed in on the keyboard.
*HELP	*H.	Displays each file system command with it's syntax.
*INFO	*I.	Displays information about specified files.
*LIB	*LIB	Selects the drive/directory for the library.
*LIST	*LIST	Displays a text file on the screen with line numbers.
*LOAD	*L.	Reads a file from disc to memory.
*OPT 1	*O.1	Switches screen messages on or off.
*OPT 4	*O.4	Specifies the auto-start option of a disc.
*RENAME	*RE.	Changes a filename.
*RUN	*R.	Runs a machine program.
*SAVE	*S.	Saves a specified part of memory to the disc.
*SPOOL	*SP.	Transfers all text subsequently displayed on the screen into a specified file.
*TITLE	*TI.	Changes the title of a disc.
*TYPE	*TY.	Displays a text file on the screen with line numbers.
*VERIFY	*V.	Checks legibility of every sector on a disc.
*WIPE	*W.	Removes specified files from the disc after confirmation for each file meeting the given specification.

A lot of people do not appreciate that a disc has to be formatted before it can be written to or read from. The BBC Drives come with a Formatting Disc and people should check that they can get hold of a Formatting Disc if they purchase drives elsewhere.

Generally, the Formatting Disc has two programs on it; one for formatting 40 track drives and the other for 80 track drives. These are usually called \*FORM40 and \*FORM80. The Formatting Disc should be write-

protected to avoid accidental erasure. This is done by covering a small notch along one edge.

Formatting prepares new discs for use with the Disc Filing System on BBC Microcomputer. It marks areas of disc where information will be stored and sets up a catalogue. The catalogue is empty at first but when you store programs and data on the disc the catalogue records their positions on the disc. They can then be retrieved quickly by reference to the catalogue. While formatting, the information put on to disc is verified automatically.

When storing files the name can only consist of seven letters as opposed to ten with the CFS.

Steve Linter

## bookreview

**Assembly Language Programming for the BBC microcomputer. I. BIRNHAM. Macmillan Computing Books, £8.95. Book and Cassettes, £16.00.**

This excellent book is written specifically for the BBC computer and it assumes an elementary knowledge of BASIC on this machine. The 300+ paged book is written in easily understood plain english and not in the more common jargon which assumes that you know the subject thoroughly as is used in most books on assembler.

The writer works his way slowly through the assembly language code and in many cases, comparing BASIC structures such as IF THEN ELSE, FOR NEXT, REPEAT UNTIL etc. with their assembly language counterparts. In each case examples and exercises are given together with answers to the exercises. There are many sections in which alternative methods are given with good explanations of their relative advantages.

The special properties and problems of the 6502 processor are well brought out and their uses described. The book not only describes how to incorporate assembly language programs into BBC BASIC but also how to drive the user ports etc.. Information is given on several of the BASIC and OS functions that I have not seen described adequately elsewhere including how to hold and use together several programs in memory at the same time and without any problems (using the PAGE function and common variables).

Detailed lists of all assembly language codes with their addressing modes and flag operations are, of course, included.

The book contains 73 listings of useful subroutines and utilities. These form the basis of the two cassettes which can also be purchased. These cassettes also each contain two other useful programs not in the book. These programs are for plotting graphs, a disassembler, a locator for any section of code and a "replace" program which replaces any code by a new code. This can be useful in changing long descriptive names to short ones when a program is working and it is desirable to shorten and speed up the program.

In conclusion, I find this book to be the most useful and best written assembly language book that I have seen and is suitable for both a beginner in assembly language programming as well as for an experienced programmer.

D.E. Susans

## oddsport

For the first time in this series, the Oddspot is written mainly in machine code. It is in actual fact a demonstration of the effects of altering the screen at close to the refresh rate. Try pressing some keys while it is running. To get out of it you will have to press BREAK. It is written by Jeremy Ruston.

```
L.
10 DIM R% 200:SZ=#7D90
20 FORT%=#TU2STEP2
30 P%=R%:LOPT TX
40 .clear:lda#255
50 jsr empty:lda#32
60 jsr empty:jmp clear
70 .empty:ldy#0
80 .emi sta SZ,Y
90 sta SZ+256,Y
100 sta SZ+512,Y
110 sta SZ+768,Y
120 iny:bne emi
130 rts:INEXT
140 MODE7:CALLclear
```

©LASERBUG 1982. Reasonable care is taken to avoid errors in this magazine. However, no liability is accepted for any mistakes which may occur. The opinions expressed in articles are those of the authors and do not necessarily express those of the Editor. No material may be reproduced in any way (for any reason) without the written consent of the Editor.

LASERBUG is edited by Paul Barbour.

The Contributors for this month were Paul Barbour, Edward Hunt, Graham Anderson, Nick Goodwin, Dr D E Susans and Steve Linter.

The final copy date for Issue 10 (March) is Wednesday 9th February. Payment for articles is at the reate of £5 or £10 per contribution depending on size and content. Other rates can be negotiated for work of high standard.

Printed by Prestige Press (UK) Ltd, East Street, Chesham, Bucks.

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

Membership: £12 for 1 year (£18 overseas), £6 for 1/2 year (£9 overseas).

Advertising rates available on request.

continued from page 21

want to go to sums greater than one million pounds!

Output from MORTGAGE could be formatted by including the function and replacing line 130 with:

```
XL 130
130 PRINTTAB(10); "YEAR"; FNformat
(2,3,0); FNformat(B,10,2)
```

Edward Hunt

continued from page 19

The next type are those that use graphics purely as a reference to where you have been i.e. as a map. The one example of this is: TIMETRAVEL – Software for All, 72 North Street, Romford, Essex. This kind of program still follows the style of the original.

Following on from this is the type that are wholly grapohics based. One such program is KINGDOM by '42' Software. This game however looses its puzzle aspect by such a style and hence makes it suitable for a different type of person. Details of this program were given in LASERBUG Issue 6, pages 11/12.

The other style of programs that can be included in the category of Adventures are completely different from the ones above and really need explaining individually.

One such program is SPACE KINGDOM by SOFTWARE FOR ALL. This is a cross between Star Trek and games such as Star Radiers (for the Atari). This is a real time space game with impressive graphics and the kind of game that makes a change from both Space Invaders and normal Adventures but unlike KINGDOM is still quite hard to beat. This game requires strategy rather than solving puzzles.

The final type we will look at is the financial type of Adventure. Every BBC owner has an example of this – KINGDOM on the Welcome tape. SIMON W HESSEL SOFTWARE produces two financial games of which GB LTD allows you to run the country and INHERITANCE sets you the task of making as much money as possible out of £10,000 that has been left to you.

The classic adventures are very good if you enjoy a puzzle. If you prefer something less taxing then perhaps one of the alternative types might take your fancy. However if you are the kind that quite happily sits for hours mindlessly zapping aliens then you have wasted your time reading this article.

If you have come up against an adventure that seems impossible to solve drop us a line and we will see if anyone else can answer your problem. Otherwise, happy adventuring . . . . .

Paul Barbour

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



SUBSCRIBE NOW TO GET THE BEST OUT OF YOUR BBC MICROCOMPUTER

Please enrol me as a Member of LASERBUG. I enclose a cheque/postal order for £12.00 for 12 months subscription.

Name .....

Address .....

.....

.....Date.....

Telephone ..... (Home) ..... (Work)

To run from issue.....

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

Please make all cheques/postal orders payable to LASERBUG ... and you can always send a copy of this form if you don't want to cut up your copy of LASERBUG...

# LASERBUG SPECIAL OFFERS

**DUST COVERS FOR YOUR BBC MICRO ONLY £3.25**  
**5% DISCOUNT ON ALL GOODS FROM DRACAL**  
**BACK COPY AND PHOTOCOPY SERVICE**  
**10% DISCOUNT ON ALL BOOKS FROM INTERFACE**

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 inclusive.

Dracal (North West) Limited 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) Limited, P.O. Box 130, Warrington, WA1 4QB. Phone Padgate 815419. Also see LASERBUG Issue 6.

Missed out on a copy of LASERBUG??? If so then you require our back copy service. Back copies are available for the price of £1.00 each and are available for all magazines printed to date except Issue 1. Brief details of the magazines are given below.

- Issue 2 (16 pages) – Sound/Envelope, \*FX Part 1, Teletext Part 2, User Definable Key & Pontoon
- Issue 3 (24 pages) – Programmers Corner, \*FX Part 2, Epson Screen Dump, Soft Review, Telesoftware
- Issue 4 (18 pages) – Hardspot, What Printer?, Bookreview, Oddspot, Pixel Power, Moving Things, Four In A Row
- Issue 5 (32 pages) – Wordprocessor, Machine Code, Disassembler, Seiksha Screen Dump, Circles Galore, Conversions for RGB Input
- Issue 6 (24 pages) – Software Protection Part I, Puzzle Program, How To Use Joysticks, Queryspot, String, String, String, What Monitor?
- Issue 7 (24 pages) – Software Protection Part II, Alphabet Tester (educational program), Speeding Up Your Programs, Computer Conversions Part I, LASERBUG Questionnaire 1982 Results, Club Reports, Business Spot

Also please either include 50p P & P for the first magazine and 25p for each subsequent one or include an A4 SAE of appropriate size and postage.

For those magazines which we have run out of stock on we offer a photocopy service. This is also available if there is a particular article or program you wish photocopied instead of buying the whole magazine. Price: 50p/article + postage (half of that for backcopies). For Issue 1 which is a 16 page magazine please send £1.60 + normal backcopy postage.

Interface have allowed us to offer you a discount on all their books. However unlike the Dracal offer you have to buy the books through us using the form below. The two books Interface offer are Making The Most Of Your BBC Micro (reviewed in Issue 4) by Tim Hartnell and The BBC Micro Revealed (reviewed in Issue 5) by Jeremy Ruston. A big thank-you to Interface for making such a kind offer.

Membership cards coming soon.

	ITEM	QUANTITY	PRICE
Please send me	LASERBUG Dust Cover(s) @ £3.25 each	.....	£ ..... p
	LASERBUG Back Copies of Issue(s) ..... @ £1.00 each	.....	£ ..... p
	LASERBUG Issue 1(s) (photocopies only) @ £1.60 each	.....	£ ..... p
	Photocopy(ies) of LASERBUG articles @ 50p/article*	.....	£ ..... p
	Copies of Making The Most Of Your BBC Micro by Tim Hartnell @ £5.81 each	.....	£ ..... p
	Copies of The BBC Micro Revealed by Jeremy Ruston @ £7.16 each	.....	£ ..... p
	Postage	.....	£ ..... p
		TOTAL ENCLOSED:	£ ..... p

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

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

PLEASE ALLOW 28 DAYS FOR DELIVERY

\*Please enclose details on a separate sheet

*(This order form supercedes all others)*