BEEBUG

Issue 2 May '82

IN THIS ISSUE:

- SOUND and ENVELOPE explained
- Operating System Commands
- User Definable Keys
- More on Teletext Graphics
- Programs

. . . . . and lots more!

2 editorial

We made it to issue two. I never really doubted that we would, it is just that I did not expect to do it in the style that we have.

Our initial response from our advertising was as we expected, but that response has greatly increased of late. Whether that has anything to do with stepped-up deliveries of the computer by BL Marketing I don't know, but one wonders. . . . .

Thanks are due to those of you who have written to us in response to issue one of the newsletter. Yes, I know, and I am sorry about the mistakes. Hopefully our teething troubles are over—at least we have got our printer modification to work (only modified by the manufacturers on issue three boards, ours is issue two) so all programs will be dumped directly onto a printer. More about the mistakes (and corrections) later.

The general concensus of your letters was: "Good work just what we want" and "Keep it up". The letters column features a few of the many letters that we have had. . . . and they were just what we needed. There is nothing like being told that you are doing just fine.

We have had several entries for our '*Program of The Month*' competition. There is no *Program of The Month* in this issue as we went to press before the closing date for the first entries. That will be one of our main spreads in the next issue. However, this month we have got an addictive game of pontoon from Ian Macaulay.

As well as letters and subscriptions, a fair number of questionnaires have been rolling in. They make for very interesting reading. We will repeat the questionnaire next month for the benfit of newer members (with slight changes to Part II), and when we have got a few more in we will attempt an analysis of your answers. That looks destined for issue four at the moment.

If you don't want to cut up your copy of the newsletter then please send a copy (photo-copy or hand-written). It breaks my heart to see ripped up pages of *LASERBUG*.

The response to our request for people to arrange local user groups has been encouraging (see *Meeting Place*), as has the number of people willing to get in touch with local users (see *Contacts*). I have been talking to the ACC (Amateur Computer Club) who are keen to help us in our venture. They already have access to a large network of varied computer clubs and as they get word of BBC based/biased/ interested groups they will let us know.

Our reviews section is a bit short on length (and content. . .) this month. Literally the day before the copy for this issue had to be at the printers two books—*Practical Programs for the BBC Computer and Acorn Atom* and *BASIC Programming on the BBC Microcomputer* came through our letterbox. Paul managed to dash off a couple of quick reivews, but nothing in-depth until next month.

We also received four games cassettes from Sinclair Software (no, not him) including *Mutant Invaders*, *Star Trek*, *Breakout* and others. Reviews of those next month.

I mentioned in last month's newsletter the position you could find yourself in if your computer went wrong on you. Since then, Acorn Computers have organised a rota of regular collections from local Service Agents to take the faulty computers back to Cambridge for repair. While this is still not an ideal situation, it is at least a step forward from the position you could have found yourself in a few weeks ago.

Service seminars have been arranged by Acorn, so minor repairs should be able to be done by the local Service Agents. Incidentlly, a copy of the first issue of the newsletter was mentioned and passed around at the introductory seminar.

It seems that the cut-price war for upgrade kits to Model B has started. I know of at least two companies who are in the market with full upgrades for less than the BBC price. It is also possible to upgrade a bit at a time as you need to (or can afford to).

Now we come to where we hang our heads in shame—last issue's mistakes. Mistakes should not appear, but almost inevitably some do, however hard you try to avoid them. Our program mistakes should be non-existent now we are dumping them directly onto a printer, and our textual mistakes should be greatly reduced now we have closer liaison with our printer.

Starting on page two, and with apologies for those we have overlooked and thanks to those of you who have pointed these ones out to us, the mistakes and corrections are:

• page 2, three lines up from the bottom of the left-hand column. Apart from the syntax being wrong, the value is different on Model A's and B's. To get a workable value type in PRINT HIMEM-TOP and add 200. Or, and neater, type in ?HIMEM-TOP+200=65. Now scroll the line a few times and repeat it with 66.

Hardware scrolling is the answer, says Mr S Cheshire. Instead of moving the contents of each byte up one line (which would take ages in a 20K screen), the processor simply tells the PAL that in future, when it sends the UHF signal to the TV, it is to put each byte one line higher. So instead of moving the content of the memory, the BBC micro moves the complete memory up the screen.

• page 4, in *Hardreview*, the modification is valid for the Model B as well.

• page 5, in *Oddspot*, line 30 should have an upper-case TO.

• page 6, in the *Labyrinth* program, the identified mistakes are:

Labyrinth 1:

```
220  IF A(R+(S-1)*H)<>0 THEN 305
490  zero in line number missing
685  IF A(A+R+(S-1)*H)=0 THEN 695
705  ...S%=S....
720  ...BPUT#B,A(I).....
730  CLOSE#B
```

Labyrinth 2:

```
40    ...A(I)=BGET#B....
50    CLOSE#B
1055  GOTO 1065
1326  FOR I=1 TO 8
3770  VDU 31,35,16,227
3820  VDU 31,I,5,225
4070  FOR I=15 TO 17
4880  VDU 31,23,8,225
4890  VDU 31,22,10,227
```

• page 9, in *Teletext Graphics*, the first paragraph under the colour table should read....PRINT CHR$(128+n)..... In the second paragraph of the second column it is stated the... the final 1 suppresses the cursor...—in fact it suppresses the prompt. The two diagrams are wrong—the figure 32 should be 64.

• page 10, line 30 of the first program should have an upper-case TO and line 90: FOR B=96 TO 127. The diagram is also incorrect as the value 32 should be 64. This will alter the lines underneath to:

```
PRINT CHR$(146)CHR$(110)CHR$(119)CHR$(61)
VDU 146,110,119,61
```

A full character set is shown in the *Teletext Graphics Part II* article in this issue.

In the second program:

```
10   colon, not =, before NEXT F
30   The 9 before the 136 mid-way through the VDU statement
     should be 19. The last pair of 117,53 should be removed.
40   FOR F=145 TO 151
45   colon, not =, before NEXT T
60   colon, not =, before NEXT G
```

• page 12, in *User Definable Characters*, all VDU statements should read VDU 23,..... then the nine parameters. In the parameters for a Pawn a 28 before the 62 is missing, and for a Rook a 42 is missing before the 28s.

As far as we can gather, that's it. If we have overlooked any, or you spot others in future issues then please let us know. Obviously we will try to keep them to a minimum, but some will slip through.

By the time you read this the IPC Computer Fair will have come and gone. We probably met a few of you at the show, but those sort of places are not the best for talking to people for any length of time. So a meeting as soon as possible will have to be arranged. Full details will be in next month's newsletter, but I can tell you it will probably be in London—time and place to be decided.

Our order of *LASERBUG* membership cards looks set to come any day now, and they will probably be sent out with your next issue of the newsletter. This will enable you to get discount on certain goods at some retailers—full details when we get everything organised.

Lastly, we owe an apology to Henry Budgett of Computing Today. The Labyrinth program we ran last month was an adaptation of a program printed in the January 1980 issue of CT, and should have been credited as such. Sorry Henry.

One more thing: our man-on-the-inside at Acorn has supplied us with some more exciting information about the BBC BASIC equivalents for PEEK and POKE, and other byte indirection operators. So the PEEK and POKE explained article is being rewritten for next month's issue of the newsletter.

That's all from me for this issue. I hope you enjoy this month's contribution as much as you seemed to enjoy last month's. Keep on writing.

Trevor Sharples

help!

I know we keep going on about how LASERBUG is your User Group and how we want to produce something that you want, but it is true. So that is why I am putting out this impassioned plea for HELP.

The LASERBUG newsletter does not write itself (as my many late nights will testify) nor do meetings get arranged by a wave of a magic wand. People have to do all those things, and at the moment there are not very many of us.

I would like to hear from people who would like to give a hand in organising the newsletter, help with arranging meetings and so on. It will be easier at first if you were local to the Romford, Essex, area as it makes it easy to keep in touch. Even if you are not local I would still like to hear from you.

Please address your letters to HELP, LASERBUG, 4 Station Bridge, Woodgrange Road, Forest Gate, London E7 0NF.

contents

user definable keys 3

One of the many things that sets the BBC micro so far ahead of its rivals is the row of ten red user-definable keys along the tope of the keyboard.

It is true that you can define keys on other micros (the VIC-20 springs to mind), but that often involves the loss of the original function of the key. Also, convoluted software adjustment is often needed to re-define the key.

The BBC micro scores over its rivals in all departments. Not only has it got ten keys especially reserved to be re-defined, it is extremely easy to define a key.

The User Guide (provisional version) explains quite well how to define a key, but it does not go on to describe how useful this can be. That is the main purpose of this article.

The syntax to define a key is very simple. The keys are numbered from 0 to 9. So to define key 0:

```
*KEY0 "INPUT"
```

Try it for yourself. Each time you press KEY0 the word INPUT is printed on the screen. It doesn't have to be INPUT—it could be anything you wanted.

So the keys can be used to contain any long words or equations that your are using repeatedly in a program. Each time you press the defined key the words or equations appear. It certainly takes a lot of effort out of typing.

As well as storing long words, we can store other commonly used commands and execute them. LIST, for example, is always useful. It is much easier to press one key to LIST a program than typing in LIST or even L. then RETURN.

The syntax for this is:

```
*KEY0 "LIST|M"
```

The funny symbol between LIST and M can be found on the key to the left of the left-cursor key. This symbol will appear as two vertical bars in MODE 7 owing to the teletext character set.

RUN is also a command that is used again and again and would benefit from being assigned to a key.

A group of schoolchidren have devised the following program which is meant to be loaded and RUN to assign functions to the user-definable keys. If you do this as soon as you switch your computer on you will find it an invaluable aid in your programming, especially KEY 8 which is a real-time clock.

Once the program has been RUN and the keys defined it can be NEWed. However, be careful when pressing BREAK. If you press it a couple of times in quick succession and get the computer to 'beep' and display the RAM size, you will reset the user-definable keys. Otherwise the keys stay defined each time BREAK is pressed.

The following program defines the key as follows:

```
KEY 0  LIST
KEY 1  RUN
KEY 2  Clear Screen
KEY 3  CHAIN " "
KEY4  RENUMBER 5,5
KEY 5  AUTO 5,5
KEY 6  * CAT
KEY 7  (unused)
KEY 8  Time
KEY 9  OLD
```

```
 5 *KEY0  |N|L|ML.|M|O|M
10 *KEY1 RUN|M
15 *KEY2 |L|M
20 *KEY3 CHAIN ""|M
25 *KEY4 REN.5,5|M
30 *KEY5 AU.5,5|M
35 *KEY6 *CAT|M
```

```
   40 *KEY9 OLD|M
   45 CLS:PRINT"InPut the time Ple
ase:"
   50 INPUT"HOURS="ho
   55 INPUT"MINS.="mi
   60 TIME=((ho*60+mi)*60)*100
   65 *KEY8 CLS:P.TAB(35,1);(TIME
DIV 360000)MOD 13;":";(TIME DIV 60
00)MOD 60|M
   70 CLS:PRINT''"Function keys as
follows:"'
   75 PRINT"f0 List"
   80 PRINT"f1 Run"
   85 PRINT"f2 Clear Screen"
   90 PRINT"f3 Chain"
   95 PRINT"f4 Renumber 5,5"
  100 PRINT"f5 Auto 5,5"
  105 PRINT"f6 *Cat"
  110 PRINT"f7 Unused"
  115 PRINT"f8 Time"
  120 PRINT"f9 Old"
```

As an aid to your memory we have printed a cut-out strip with the user-definable key functions on it (plus some spares for your own use). This strip goes underneath the clear plastic band above the red keys. These strips are on the back page of this newsletter.

Although there are ten red keys to be defined, it seems that the men at Acorn were not satisfied. The BREAK key can also be defined. BREAK is key 10—so *KEY10 "MODE 5:M" will automatically put you into MODE 5 everytime BREAK is pressed.

We have heard rumours that the cursor control keys and COPY can also be defined, but we cannot get them to work (even using *FX4, 1 to disenable them). More about them when we are successful.

teletext graphics part II

Our article on Teletext Graphics in issue one of the *LASERBUG* newsletter seemed to catch a lot of people's attention. Your comments ranged from "Excellent stuff" to ". . . looks good but I don't understand it".

As an article it was only intended as an introduction to the graphical capabilities of the BBC micro in MODE 7, with a few examples of how to obtain colour and how to generate the graphics.

Dave Leedham has taken things one step further and he has written a program to function as a simple Teletext page editor. This program allows you to easily explore the Teletext graphics and to design graphical screens for later inclusion in programs.

The associated diagram provides the complete set of alphanumeric characters, graphical characters, control characters and control codes. The diagram is a modified version of the Prestel Transmission Codes taking into account the key symbols on the BBC micro. This diagram should be read in conjunction with the instructions for the Teletext Editor.

The listing for the program is suprisingly short:

```
    5 REM*************************
*
   10 REM**** Teletext Editor  ***
*
   20 REM**** D.C.Leedham 1982 ***
*
   25 REM*************************
*
   30 ON ERROR GOTO120
   40 MODE 7:*FX4,1
   50 G=0
   60 C=GET
   70 IF C>135 AND C<140 THEN C=C-
128:VDUC:GOTO60
   80 IF C=9 THEN G=64:C=GET
   90 IF C=127 THEN P=255 ELSE P=C
+G
  100 VDUP
  110 GOTO50
  120 *FX4,0
```

When the program is RUN the screen should clear and the cursor will appear in the top left-hand corner. The four cursor control keys will allow you to move the cursor around to any point on the screen.

Any characters typed in will appear at the present cursor position in Alphanumeric White (that is as usual in MODE 7) unless a control character has been entered earlier on the line.

To enter a control character on the screen press the TAB key followed by the desired code. The control codes can be found in columns four and five of the diagram. For example: to get a line of red alphanumeric characters press TAB and then press A. Any further upper-case letters entered on that line will appear in red.

It should be remembered that the control codes only affect the line they are entered on. A further control code will have to be entered on the following line to achieve any affect other than alphanumeric white.

Further examples of control codes are: TAB then H will give you flashing characters (TAB then I to stop them flashing); TAB then S will give you yellow graphics and TAB then Z will give you separated (as opposed to contiguous) graphics.

When using control codes to get graphical characters, the lower-case letters, numbers and special symbols will print out the graphical character set (see columns two, three, six and seven of the diagram).

It is possible to use a combination of more than one control code. For example: TAB then Z then TAB then T then TAB then H will give you flashing blue separated graphical characters. However, you should bear in mind that each control code is printed as a space, and if a control code is overwritten it becomes nullified.

Double-height characters can be obtained by pressing TAB then M, typing the required phrase, moving the cursor to the line below, pressing TAB then M again and then the same phrase to complete the bottom half of the characters.

To obtain the graphical character &7F/128 (the inverse space) use the DELETE key.

To erase any character you have printed place the cursor below the character and press the space bar.

If you want to leave the program press ESCAPE. You may need to press ESCAPE twice if you are not in the alphanumeric white mode.

Dave's program will give you an excellent opportunity to experiment with Teletext graphics, and you can end up with some quite sophisticated designs indeed. If you discover anything unusual, or if you generate any interesting designs, then please let us know.

NOTE:

Columns 0 and 1 form the C0 control character set.

Columns 4b and 5b form the C1 set of display attribute control codes.

Columns 2, 3, 4, 5, 6 and 7 form the G0 character set.

Columns 2a, 3a, 4, 5, 6a and 7a form the graphics character set.

Column 3b is a non-standardised set of control characters, used from computer to terminal only.

*From terminal to computer only.

b_7	0	0	0	0	1	1	1	1
b_6	0	0	1	1	0	0	1	1
b_5	0	1	0	1	0	1	0	1

Bits b_7 b_6 b_5 b_4 b_3 b_2 b_1	Col / Row	0	1	2	2a	3	3a	3b	4	4b	5	5b	6	6a	7	7a
0 0 0 0	0	NUL				0			a		P		–		p	
0 0 0 1	1		Cursor on	!		1		Set verify mode (1)	A	Alphanumeric Red	Q	Graphics Red	a		q	
0 0 1 0	2			"		2		Set verify mode (2)	B	Alphanumeric Green	R	Graphics Green	b		r	
0 0 1 1	3			£		3		Skip block	C	Alphanumeric Yellow	S	Graphics Yellow	c		s	
0 1 0 0	4		Cursor off	$		4		Set programme mode	D	Alphanumeric Blue	T	Graphics Blue	d		t	
0 1 0 1	5	ENQ		%		5		Tape pause on playback	E	Alphanumeric Magenta	U	Graphics Magenta	e		u	
0 1 1 0	6			&		6		Tape start	F	Alphanumeric Cyan	V	Graphics Cyan	f		v	
0 1 1 1	7			'		7		Tape stop	G	Alphanumeric White	W	Graphics White	g		w	
1 0 0 0	8	Cursor left	Delete line*	(		8			H	Flash	X	Conceal Display	h		x	
1 0 0 1	9	Cursor right		)		9			I	Steady	Y	Contiguous Graphics	i		y	
1 0 1 0	10	Cursor down		*		:			J	End edit*	Z	Separated Graphics	j		z	
1 0 1 1	11	Cursor up	ESC	+		;			K	Start edit*	←	[	k		¼	
1 1 0 0	12	Cursor home & clear screen	Single shift 2	,		<			L	Normal Height	½	Black Background	l		‖	
1 1 0 1	13	Cursor return	Single shift 3	–		=			M	Double Height	→	New Background	m		¾	
1 1 1 0	14	Shift out	Cursor home	.		>			N	Start insert	↑	Hold Graphics	n		÷	
1 1 1 1	15	Shift in		/		?			O			Release Graphics	o		DEL	

6 letters

LASERBUG's *post is growing daily. We have been surprised, and pleasantly so, by the interest that a BBC microcomputer User Group has generated. It seems that many BBC micro owners are determined to get as much out of their computer as they can, and they see a User Group as being the best way of achieving that objective. We like to think that you are right, because that too is our belief. Mr G. Duplock of Littlebourne perfectly summarises so many of your letters:*

"Having recently received my BBC micro I read your advertisement in Your Computer with great interest. I fully support the idea of an independent User Group to circulate information, offer support to users and, if possible, apply pressure to enhance and direct the BBC in their micro adventure. Looking forward to hearing from you."

That was by no means the only letter we received along those lines. Those of you who had seen issue one of the LASERBUG *newsletter were just as enthusiastic about us. We have had several letters of congratulation, like this one from Nick Lamb of Great Yarmouth:*

"Thanks for a very good first newsletter. The articles about the Teletext graphics and user definable characters were a perfect start – just the thing to help us get the most out of our micros (when they arrive). I'm very enthusiastic about the BBC system and LASERBUG. Thanks for getting it off to a flying start."

Thanks Nick, we are glad you enjoyed the newsletter. We only hope we can keep getting better issue by issue.

It is nice to know that the computer magazine giants (PCW, PC, YC, CT, E & CM, C & VG. . . .) do not always look down with scorn on User Group newsletters like ours. Henry Budgett, Editor of Computing Today, had some good things to say about us:

computing today

Published by Argus Specialist Publications Ltd
145 Charing Cross Road, London WC2H 0EE Tel (01) 437 1002 Telex No 8811896

Tuesday 6th April 1982

Dear Trevor

Thank you for the first issue of Lazerbug, a very impressive production indeed. I hope that you can keep up the good work for a long time to come...decently produced newsletters seem to be a rarity these days.

On a completely different topic I would like to ask that you consider using CT's standards for graphic presentation, a number of publishers are already using it and a book on the BBC Micro to be published shortly will be based on it. If you'd like to join the growing band of standardised publications drop me a line or come and see us and we'll fill you in on the details.

Yours sincerely

Henry Budgett

Registered Office: 12-18 Paul Street, London EC2. Registered in England No. 1269520
Directors: T. Gold Blyth (Chairman), T.J. Connell (Managing), E. Hurworth, A.T. Patten, N.E. Richards

We certainly intend to keep up the good work, and thanks to Henry for his encouragement. Thanks are also due to all of you who have written to us with letters of encouragement. Peter Hughes of Bristol for instance:

"Many congratulations on your first issue of LASERBUG. A nicely balanced issue considering the short time lag from advert to print production.

P.S. Congratulations on a clever name."

Well – we like it. The production of the first issue of any magazine/newsletter is always hectic, and it usually takes two or three issues to arrive at a suitable format. So please bear with us as we try to get things right. Constructive criticism is always welcome.

Mr J. Reid precied his praise down into two sentences:

"I was most impressed reading issue one of the LASERBUG newsletter. Best of luck with the endeavour."

Chris Drage, a veteran of the letters page of issue one of the newsletter, has written to us with more than glowing words of praise. His letter should be especially interesting for those of you who are considering the purchase of a colour monitor for your computer:

"Thanks for a very fine first edition of LASERBUG – well done. Thank you, also, for the answer to my cassette recorder query.

Readers may be interested to learn that two London companies – Portatel of Sunbury and Lux Computers of Watford – are currently fitting RGB inputs to two television sets – the SALORA and LUXOR respectively. I have been considering purchasing such a modified TV/Monitor as the purchase of a colour monitor is a considerable outlay for a piece of equipment which has limited use. By paying another £60–£70 one has a computer monitor on one hand and the benefits of a portable colour television on the other.

Both companies tell me that the results on an Apple II are excellent – full monitor clarity and crispness with none of the problems of UHF output. It still remains to be seen if similar results can be achieved with a BBC micro in MODE 0.

Keep up the good work."

If anyone tries the BBC micro on one of these sets then please let us know how you get on. We are a little bit dubious as to the claims of the manufacturers, but we would like to be convinced. There is a great market for good, cheap colour monitors, and as Chris says, it is worth paying a bit extra to get the additional benefits of a colour television.

As well as receiving letters saying good things about LASERBUG, *it is refreshing to read letters with good words about the BBC micro itself – even if things don't always go right. Ian MacAulay of Shutlanger takes up the story. . . .*

"Thanks for the first issue of the LASERBUG newsletter. I was very impressed with it. I hope future editions will be just as good. I thought that I'd let you know of my experiences with my BBC micro.

I ordered a Model A machine on 5 November 1981 after serious consideration of rival micros such as the VIC 20, Texas and Tandy.

The micro arrived on 17 February 1982. Initially I was a bit disappointed – the lid of the UHF modulator was lying loose on the main circuit board, the space bar was jammed and I didn't think much of the 'flexible' plastic casing with all the various holes and gaps. I wasn't impressed with the thin User Guide and there was a complete absence of guarantees or warranties.

However, I really enjoyed my first efforts. I now believe the BBC micro to be easily the best on the market in the hobbyist's price range.

I think the BASIC is superb – I particularly like the PROCs – the assembler looks promising and the keyboard is very good for a small micro. The screen editor is a delight compared to the PET's. At first the graphics seem very good, but they do use large amounts of memory.

Unfortunately I don't have enough free time to make the best use of my micro, but overall I am very impressed with it. I now look forward to obtaining software, peripherals etc., and I look forward to the next issues of your newsletter."

Another satisfied customer. Mr P. Thompson of Godmanchester, Cambs., also has a Model A and a dilemma with which the Editor deeply sympathises:

"I have a Model A BBC micro. It seems to be bug-free and reliable, even after many hours of continuous running. I will now be reluctantly selling a faithful Atom and seeking the info on increasing the BBC's RAM up to 32K."

Sad news. Although Acorn Computers are now advertising a ROM upgrade kit to allow the ATOM to use BBC BASIC. Obviously, some of the hardware-dependent functions and the higher resolution graphics won't be available, but it does mean that the ATOM is not dead.

Paul Chand asks a question along a different line:

"Would you be able to tell me whether it will be possible to use the BBC micro with a ZX81 printer to get lisings printed?"

Quicksliva – ZX81 hardware specialists – have already managed to run a ZX81 printer off an Acorn ATOM, and they tell us that the BBC micro is their next project. Watch this space.

Lastly, Mr R. Andrews has a word of advice for the first-time owner of a BBC micro who is put off by the lack of a plug on the far end of the BBC supplied cassette lead:

"Buy a new five pin DIN to five pin DIN straight wired lead in any shop dealing with Hi-Fi goods (e.g. £1.40 in W. H. Smiths). Don't worry about the BBC micro having a seven pin socket. Plug it into a standard five pin socket in any common cassette recorder, and SAVEing and LOADing programs will be no problem. Motor control is, of course, a little more difficult!"

. . . . . and it works. It also saves a lot of hassle soldering up the correct plug.

Well, that's all from the Mailbag for this issue. More missives next month. Keep the letters coming.

oddspot

Last month's *Oddspot* certainly caught a few people out. It was definitely unfair on those of you who had not yet received your computer, but even so, several of you worked out exactly what was supposed to be going on.

For those of you that are still in the dark, last month's program drew a series of thin horizontal lines of random colours (some flashing) that gradually moved up the screen. Not particularly exciting, but intriguing.

This month, *The Phantom* of Mill Hill challenges you to work out *exactly* what you will see on the screen when you run this program.

```
10 MODE4
20 VDU5
30 CLS
40 X=1 : Y=1
50 MOVEX,Y
60 PRINT"LASERBUG"
70 X=X+10 : Y=Y+10
80 IFX=891  THENX=1 : Y=1 : CLG
90 GOTO50
```

8 pontoon

Ian MacCaulay of Shutlanger, Northants, reveals his gambling inclination in this game of Pontoon (or Blackjack, Twenty-one or Vingt-et-un). The program is written in under 16K so it will RUN on both the Model A and B BBC microcomputer.

Rules of the Game:

This program is based on the popular card game of Pontoon. The basic idea is for each player to try to make a score of near to 21 as possible without exceeding that figure, and then the dealer tries to beat the players.

The values of the cards are as follows:

2-9	face value
10,J,Q,K	10 points
Ace	1 or 11 points

The Program:

On RUNning the program you are first asked to input the number of players taking part. Between one and five players can play. The computer acts as the dealer and handles all the cards and the betting—without cheating.

Initially, each player receives 200 units (pence or whatever) and the computer (player 0) receives 1000 units. The dealer then puts 6 units into the pot.

Once the number of players is input the current score is displayed. Press the SPACE bar to continue.

The first player is dealt a card, and the program then waits for a bet. This may be between one and the value of the pot. (The pot is displayed in the centre of the screen near the top).

The cards are represented by two characters—the left-hand one is the value (2-9, 1=Ten, J=Jack, Q=Queen, K=King) and the right-hand one is the suit (S=Spades, H=Hearts, D=Diamonds, C=Clubs).

Once a valid bet has been input the second card is dealt. The computer will now wait for instructions.

—if you have 16-21 points you may STICK by pressing S <RTN>.

—if you have 13 or 14 points you may BURN by pressing B <RTN>. Your two cards are then replaced by another pair and your bet is doubled (up to the maximum of the pot).

—if you have 15 points you may DEALER'S HAND by pressing D <RTN>. Your two cards are now exchanged for the pair held by the dealer, and the dealer gets your pair. Your bet is also doubled.

—on any score you may TWIST by pressing T <RTN>. You are then dealt another card.

If you exceed 21 you BUST and the pot gets your bet and the game moves on to the next player. Once you have STUCK the dealer then reveals its hand.

Press the SPACE bar.

The dealer will either STICK, TWIST or BURN. After each action by the dealer press the SPACE bar to continue—this is to allow you to follow the progress of the game. Once the dealer has STUCK the two hands are compared to judge the winner. The hierarchy or scoring is:

Pontoon (Ace + King, Queen or Jack)
5 Card Trick (5 cards of any value points < 21)
Junior Pontoon (Ace + 10)
21 points
20 points—down to 16 points

If both hands are the same then the dealer wins. If the dealer wins then your bet is put into the pot.

If the player wins then the dealer's bet is put onto the player's score from the pot.

Press the SPACE bar to continue. The current score is displayed. Press the SPACE bar again to move on to the next player.

If the pot goes down to zero then it is refilled to six from the dealer's score.

If the pot survives two rounds then all but six is emptied into the dealer's score.

The game continues until the ESCAPE key is pressed as the cards are replaced on the bottom of the deck. The LASERBUG play-test team amassed a fortune of 291 units. Can you beat that?

```
    1 MODE 7
    3 PRINT TAB(0,6)CHR$(141)"PONT
OON"
    4 PRINT TAB(0,7)CHR$(141)"PONT
OON"
    6 PRINT TAB(1,8);"_______"
    7 PRINT TAB(0,12)"   by"
    8 PRINT TAB(0,15)"I.A.MacAula
y"
   10 DIM NACE(5),NC(5),NPT(5,5),A
(52)
   20 DIM C$(5,5),A$(52),B$(52)
   25 DIM M(5),NKING(5),NTEN(5),NP
ON(5),NJPON(5)
   30 FOR I=1 TO 52
   40   READ A$(I)
   45   A(I)=0
   50   NEXT I
   60 DATA 1S,2S,3S,4S,5S,6S,7S,8S
,9S,JS,QS,KS,AS
   70 DATA 1H,2H,3H,4H,5H,6H,7H,8H
,9H,JH,QH,KH,AH
   80 DATA 1D,2D,3D,4D,5D,6D,7D,8D
,9D,JD,QD,KD,AD
   90 DATA 1C,2C,3C,4C,5C,6C,7C,8C
,9C,JC,QC,KC,AC
  100 FOR I=1 TO 52
  110   X=RND(52)
  120   IF A(X)=0 THEN 140
  130   X=X+1
  135   IF X=53 THEN X=1
  137   GOTO120
  140   A(X)=I
  150   NEXT I
  160 FOR I=1 TO 52
  170   B$(I)=A$(A(I))
  180   NEXT I:IC=52
  190 CLS
  200 PRINT TAB(0,15)"Number of Pl
ayers ";
  205 ND=0
  210 INPUT" "NP
  220 IF NP>5 OR NP<1 THEN 190
  222 FOR I=1 TO NP
  224   M(I)=200
  226   NEXT I
  228 M(0)=1000
  229  POT=6:M(0)=M(0)-6
  230 X=0:IF ND=2 THEN M(0)=M(0)+P
OT:POT=6:M(0)=M(0)-6:ND=0
  240 X=X+1:IF X>NP THEN ND=ND+1:G
OTO 230
  245 PROCMON
  246 PROCCLR
  250 CLS
  260 PRINT TAB(0,4)"PLAYER ";X
  265 PRINT TAB(13,6);POT
  270 PROCDEAL(X)
  280 PROCDEAL(0)
```

```
  290 PRINT TAB(0,8)C$(X,1)
  300 PRINT TAB(0,6)" ";:INPUT" "S
T
  305 IFST<1 OR ST >POT THEN PRINT
 TAB(0,6)"          ":GOTO 300
  310 PROCDEAL(X):PROCDEAL(0)
  320 PRINT TAB(0,8)C$(X,1):PRINT
TAB(0,9)C$(X,2)
  330 PRINT TAB(0,20)"          "
  340 PRINT TAB(0,20)" ";:INPUT" "
COM$
  350 IF COM$="S" THEN 400
  360 IF COM$="T" THEN 450
  370 IF COM$="B" THEN 510
  380 IF COM$="D" THEN 610
  390 GOTO 330
  400 PROCPTS(X)
  405 IF NC(X)=5 THEN 1000
  410 IF NPT(X,0)>15 AND NPT(X,0)<
22 THEN 1000
  420 IF NPT(X,1)>15 AND NPT(X,1)<
22 THEN 1000
  430 IF NPT(X,2)>15 AND NPT(X,2)<
22 THEN 1000
  440 GOTO 330
  445 PRINT TAB(0,20)"TWIST   "
  450 IF NC(X)=5 THEN 330
  451 PRINT TAB(0,20)"TWIST   "
  453 PROCDEAL(X):PRINT TAB(0,7+NC
(X))C$(X,NC(X))
  460 PROCPTS(X)
  470 IF NPT(X,0)>21 AND NPT(X,1)>
21 AND NPT(X,2)>21 THEN 490
  480 GOTO 330
  490 PRINT TAB(0,20)"BUST    "
  493 M(X)=M(X)-ST
  495 IF GET$<>" "THEN 495
  497  POT=POT+ST
  500 PROCREP(X):PROCREP(0):GOTO 2
40
  510 IF NC(X)<>2 THEN 330
  520 PROCPTS(X)
  530 IF NPT(X,0)=13 OR NPT(X,0)=1
4 THEN 570
  540 IF NPT(X,1)=13 OR NPT(X,1)=1
4 THEN 570
  550 IF NPT(X,2)=13 OR NPT(X,2)=1
4 THEN 570
  560 GOTO 330
  570 PROCREP(X)
  575 PRINT TAB(0,20)"BURN    "
  580 PROCDEAL(X)
  590 PROCDEAL(X)
  595 PROCPTS(X)
  597 PRINT TAB(0,8)C$(V,1):PRINT
TAB(0,9)C$(V,2)
  600 ST=ST*2:IF ST>POT THEN ST=PO
T
  605 PRINT TAB(2,6);ST:GOTO 320
  610 IF NC(X)<>2 THEN 330
  620 PROCPTS(X)
  630 IF NPT(X,0)=15 OR NPT(X,1)=1
5 OR NPT(X,2)=15 THEN 650
  640 GOTO 330
  650 Z1$=C$(0,1):Z2$=C$(0,2)
  655 PRINT TAB(0,20)"DEALERS"
  660 C$(0,1)=C$(X,1):C$(0,2)=C$(X
,2)
  670 C$(X,1)=Z1$:C$(X,2)=Z2$
  680 ST=ST*2:IF ST>POT THEN ST=PO
T
  690 PRINT TAB(2,6);ST
  700 GOTO 320
 1000 PRINT TAB(0,20)"STICK":PRINT
 TAB(20,4)"DEALER"
 1005 ST1=ST
 1010 PRINT TAB(20,8)C$(0,1)
 1020 PRINT TAB(20,9)C$(0,2)
 1030 IF GET$<>" "THEN 1030
 1035 PRINT TAB(22,6);ST1
 1040 PROCPTS(0)
 1045 IF NPT(0,0)>21 AND NPT(0,1)>
21 AND NPT(0,2)>21 THEN 1400
 1050 IF NPT(0,0)>16 AND NPT(0,0)<
22 THEN 1260
 1060 IF NPT(0,1)>16 AND NPT(0,1)<
22 THEN 1260
 1070 IF NPT(0,2)>16 AND NPT(0,2)<
22 THEN 1260
 1080 IF NPT(0,0)=13 OR NPT(0,0)=1
4 AND NC(0)=2 THEN 1170
 1090 IF NPT(0,1)=13 OR NPT(0,1)=1
4 AND NC(0)=2 THEN 1170
 1100 IF NPT(0,2)=13 OR NPT(0,2)=1
4 AND NC(0)=2 THEN 1170
 1110 IF NPT(0,0)=16 AND RND(1)>0.
75 THEN 1260
 1120 PRINT TAB(20,20);"TWIST
"
 1130 PROCDEAL(0)
 1140 PRINT TAB(20,7+NC(0));C$(0,N
C(0))
 1150 GOTO 1030
 1170  IF NC(0)<>2 THEN 1120
 1175 PRINT TAB(20,20)"BURN    "
 1180 ST1=ST1*2:IF ST1>POT THEN ST
1=POT
 1190 PRINT TAB(22,6);ST1
 1200 PROCREP(0)
 1210 PROCDEAL(0)
 1220 PROCDEAL(0)
 1230 GOTO 1010
 1260 IF NC(0)=2 AND NPT(0,0)=21 T
HEN 1280
 1270 IF NC(X)=5 AND NC(0)<5 THEN
1120
 1280 PRINT TAB(20,20)"STICK"
 1290 IF GET$<>" "THEN 1290
```

```
 1291 PROCPTS(X):PROCPTS(0)
 1293 GOTO1600
 1295 GOTO1300
 1300 PROCREP(0)
 1310 PROCREP(X)
 1315 IF POT=0 THEN POT=6:M(0)=M(0
)-6
 1320 GOTO 240
 1400 PRINT TAB(20,20)"BUST      "
 1405  GOTO 1700
 1410 IF GET$<>" "THEN 1410
 1420 GOTO 1290
 1600 IF NPON(0)=1 THEN 1800
 1610 IF NC(0)=5 AND NPON(X)=0 THE
N 1800
 1620  IF NC(X)=5 OR NPON(X)=1 THE
N 1700
 1630 IF NJPON(0)=1 THEN 1800
 1640 IF NJPON(X)=1 THEN 1700
 1650 N1=0:N0=0
 1655 FOR I= 0 TO 2
 1660   IF NPT(0,I)>N0 AND NPT(0,I
)<22 THEN N0=NPT(0,I)
 1665   IF NPT(X,I)>N1 AND NPT(X,I
)<22 THEN N1=NPT(X,I)
 1670   NEXT I
 1675 IF N1>N0 THEN 1700
 1680 GOTO 1800
 1700 PRINT TAB(10,22)"PLAYER ";X;
" WINS"
 1710 M(X)=M(X)+ST1
 1713 POT=POT-ST1:IF POT<0 THEN PO
T=6:M(0)=M(0)-6
 1715 IF GET$<>" "THEN 1715
 1720 GOTO 1295
 1800 PRINT TAB(10,22)"DEALER WINS
    "
 1810 POT=POT+ST
 1812 M(X)=M(X)-ST
 1815 IF GET$<>" "THEN 1815
 1820 GOTO 1295
 3000 DEF PROCDEAL(V)
 3010 C$(V,NC(V)+1)=B$(1)
 3020 NC(V)=NC(V)+1
 3030 FOR I=1 TO 52
 3040   A$(I)=B$(I)
 3050   NEXT I
 3060 FOR I=1 TO IC-1
 3070   B$(I)=A$(I+1)
 3080   NEXT I
 3090 B$(IC)="   "
 3100 IC=IC-1
 3110 ENDPROC
 3120  DEF PROCREP(V)
 3130 FOR I=1 TO NC(V)
 3140    B$(IC+1)=C$(V,I)
 3145   C$(V,I)=" "
 3150   IC=IC+1
 3160   NEXT I
 3170 NC(V)=0
 3180 ENDPROC
 3200 DEF PROCPTS(V)
 3210 NPT(V,0)=0:NPT(V,1)=0:NPT(V,
2)=0:NACE(V)=0
 3215 XXX=V
 3220 FOR I=1 TO NC(V)
 3230   Z$=LEFT$(C$(V,I),1)
 3240   IF Z$="J" OR Z$="Q" OR Z$=
"K" THEN NPT(V,0)=NPT(V,0)+10:NPT(
V,1)=NPT(V,1)+10:NPT(V,2)=NPT(V,2)
+10:NKING(V)=NKING(V)+1:GOTO 3270
 3245   IF Z$="1" THEN NTEN(V)=NTE
N(V)+1:NPT(V,0)=NPT(V,0)+10:NPT(V,
1)=NPT(V,1)+10:NPT(V,2)=NPT(V,2)+1
0:GOTO 3270
 3250   IF Z$="A" THEN NACE(V)=NAC
E(V)+1:GOTO 3270
 3260   Z=VAL(Z$):NPT(V,0)=NPT(V,0
)+Z:NPT(V,1)=NPT(V,1)+Z:NPT(V,2)=N
PT(V,2)+Z
 3270   NEXT I
 3280 IF NACE(V)=1 THEN NPT(V,0)=N
PT(V,0)+11:NPT(V,1)=NPT(V,1)+1:NPT
(V,2)=NPT(V,0)
 3290 IF NACE(V)=2 THEN NPT(V,0)=2
+NPT(V,0):NPT(V,1)=12+NPT(V,1):NPT
(V,2)=22
 3300 IF NACE(V)=1 AND NKING(V)=1
AND NC(V)=2 THEN NPON(V)=1
 3400 IF NACE(V)=1 AND NTEN(V)=1 A
ND NC(V)=2 THEN NJPON(V)=1
 3500 ENDPROC
 4000 DEF PROCMON
 4010 CLS
 4020 FOR I=0 TO NP
 4030   PRINT TAB(0,2*I+6)"PLAYER
";I;"     "M(I)
 4035   NEXT I
 4037 PRINT TAB(0,2*I+8)"   POT
   ";POT
 4040 IF GET$<>" "THEN 4040
 4050 ENDPROC
 5000 DEF PROCCLR
 5010 FOR I=0 TO 2
 5020   NPT(0,I)=0
 5030   NEXT I
 5040 T=X-1:IF T=0 THEN T=NP
 5050 FOR I=0 TO 2
 5060   NPT(T,I)=0
 5070   NEXT I
 5080 FOR I=0 TO NP
 5090   NACE(I)=0
 5100   NKING(I)=0
 5110   NTEN(I)=0
 5120   NPON(I)=0
 5130   NJPON(I)=0
 5140   NEXT I
 5150 ENDPROC
```

redefined characters revisted

Paul Firmin of Chelmsford rose to the bait at the end of the User Definable Characters article in issue one of the newsletter. Within days of printing the newsletter he had sent in a rather neat program to take all the effort out of working out the decimal equivalents of the binary rows. Over to Paul to explain how the program works.

Following on from your *User Definable Characters* article in LASERBUG issue one, you may be interested in the short program below which is a useful routine for determining the decimal values required for character definition using the VDU 23 command.

Upon RUNning the program, all you need to do is type in the binary series of 1s and 0s representing each separate single row of the character you wish to define. Hit RETURN and the computer will automatically calculate and display the decimal equivalent.

This program may be condensed, represented as it is here for the purpose of clarity only. You could even place it in a program that you are writing, RUNning the special routine as and when necessary for generating special character data. This allows you to RUN and re-RUN the main program to check and change characters defined and see what they look like as you go.

To execute the routine within the body of another program just type and enter GOTO 1000—you will have to press the ESCAPE key to get out of it though.

```
 1000 RESTORE1090
 1010 T=0
 1020  PRINT"InPut binary ";:INPUT
A$
 1030 FORI=1 TO LEN(A$)
 1040   READ F
 1050   C$=MID$(A$,I,1)
 1060   IF C$="1" THEN T=T+F
 1070   NEXTI
 1080 PRINT"Decimal = "T
 1090 DATA 128,64,32,16,8,4,2,1
 1100 GOTO1000
```

As a footnote to Paul's article, a whisper on the grapevine tells us that the memory map on the 1.0 series Operating Systems is slightly different to that on the 0.10 Operating System. All BBC Micro supplied, at least up until the end of April, will have the 0.10 Operating System. To check what yours is, enter *FX0 as a direct command. You should see 'OS EPROM 0.10', or something very similar.

If what we hear is true, they are changing the location for the storage of the data for the characters (except 224 to 255). This means that the statement PAGE = 4096 will have to be changed to a different number. However, we shall have to wait and see.....

12 sounds

One question that is raised in several of your letters is: "Do you know the syntax for the SOUND and ENVELOPE commands?" The answer is yes. Rather than answer each letter individually we have decided to explain it in full as an article in LASERBUG.

In the last issue of the LASERBUG newsletter we briefly explained one way of using the SOUND statement. However, there is much more to it than that. For those would-be composers who are not familiar with buffer-flushes and synchro starts-here goes.

The SOUND statement has four parameters separated by commas:

1) The first parameter in the SOUND statement can be represented as & HSFC. The letters represent:

H=0—produce sound (optional—the statement defaults H to zero if not specified).
H=1—silent
S—number of sounds to be held up to give a synchro-start chord. Zero if no chord required or two for a three note chord.
F=0—add the sound to the queue.
F=1—flush the sound channel's buffer.
C—if one, two or three gives the sound channel: if zero gives the white noise channel.

2) The second parameter, if negative, gives the volume between the limits of −15 (loudest) to −1 (softest). If the second parameter is postive it indicates the corresponding ENVELOPE number (one, two, three or four).
3) The third parameter, if C of parameter one is one, two or three, is the pitch of the sound between the limits 1 (very low) to 255 (very high). If C of parameter one is zero (i.e. the white noise channel) then the third parameter becomes the noise number of the noise to be produced.
4) The fourth parameter is the duration of the sound.

If all the above sounds a bit daunting then an example should clear matters up. If we try to produce a three note chord:

```
SOUND &0211,-15, 30, 50
```

We are producing a sound (although you will not hear anything yet) with H set to zero (*produce sound*), S set to two (*hold up two sounds before playing chord*), F set to one (*flush sound channel's buffers*) and C set to one (*sound channel one*). The volume is set to −15 (*loudest*), the pitch to 30 with a duration of 50.

We can now add the second note of the chord:

```
SOUND &0212,-15,60,50
```

Still no sound has been produced. Not until the third note of the chord has been entered and RETURN pressed will you hear the chord. The third note is:

```
SOUND &0213,-15,90,50
```

That was worth waiting for. However, the SOUND statement used by itself produces not very exciting sounds. That is where the ENVELOPE statement comes in.

The ENVELOPE statement is a bit trickier to use as it has 14 parameters. The parameters are:

ENVELOPE
N,S,PS1,PS2,PS3,NS1,NS2,NS3,AR,DR,SR,RR,PL,SL

The parameters are defined as:

N—the ENVELOPE number (1-4) corresponding to the defined second parameter of any associated SOUND statements.
S—the duration of each time step.
PS1,2,3,—the amount to be added to the pitch in each time step.
NS1,2,3,—the number of steps for which each pitch step is to be added. For example, if PS1 is −2 and NS1 is 30, then for the first 30 time steps two will be subracted from the tone. Then PS2 will be added for NS2 time steps, then PS3 for NS3 and then around to PS1 again.
AR—the attack rate is a value added to the volume in each time step until the volume reaches the PL (peak level). The AR must be a positive value.
DR—the decay rate is a negative value determining the drop in volume after the PL is reached. This is effective until the volume drops below the SL (sustain level).
SR—the sustain rate works in a similar fashion to the DR except that it keeps operating until the end of the duration specified in the associated SOUND statement(s).
RR—the release rate is a negative value that, providing there are no more sounds waiting to be produced in the sound queue, any remaining volume is decayed away at.
PL—the peak level (volume) must be in the range zero (silent) to 126 (loudest).
SL—the sustain level falls in the range zero to PL (whatever was decided for the peak level).

With the great variety of parameters at your fingertips you can create some really interesting sounds from Space Invader effects to the full range of electronic organ/synthesiser sounds.

The following program is only a simple demonstration of what can be achieved using SOUND and ENVELOPE together. Experiment with these statements and see what you can come up with.

Our thanks go to Mr S Cheshire for his help in providing some of the information for this article.

2 REPEAT

```
  5   CLS
 10   PRINT'"Which noise ?";:R=V
AL(GET$)
 12    IF R>3 OR R<1 THEN 10
 15   PROCnoise(R)
 20    UNTIL0
 90 END
 95 DEF PROCnoise(n)
100  ON n GOTO 110,115,120
110 ENVELOPE 1,1,5,-5,-5,250,25,
25,0,0,0,1,126,126:SOUND 1,1,0,100
:ENDPROC
115 ENVELOPE 2,1,20,-15,-15,6,3,
3,127,0,-1,-1,126,126:SOUND 2,2,10
0,100:ENDPROC
120 PROCnoise(1):PROCnoise(2):EN
DPROC
```

*FX

The BBC microcomputer includes a large and powerful operating system. This can be easily accessed from BASIC and enables you to do several things that BASIC is incapable of doing. If a BASIC statement starts with an asterisk then the whole of the rest of the statement is passed to the operating system.

There are several operating system commands. Those of you familiar with the Acorn Atom will recognise some of them, but one of the most interesting is the *FX range of statements—and that is what this article is concerned with.

Not all of the *FX statements will work on the early BBC micros fitted with the 0.10 EPROM operating system (OS). As, at the time of printing, no-one has received a BBC micro with the 1.0 OS we will only list those *FX statements that work on the 0.10 OS.

The *FX statement is used to control a large number of the computer's 'special effects' from the rate of flash of the flashing colours to selecting serial or parallel printer interfaces.

All *FX calls are available in machine code. If the BASIC form of a *FX statement is:

```
*FX 137,0,1
```

In assembly language the form would be:

```
LDA #137
LDX #0
LDY #1
JSR &FFF4
```

All *FX statements greater than 126 can only be used in assembly language programs, and not from BASIC.

The *FX statements available on the 0.10 OS in BASIC are:

*FX0—this will print a message giving the version number of the operating system (0.10 in most cases. Please let us know if you have something different).

*FX4,0—will enable COPY and the cursor control keys (if previously disenabled).

*FX4,1—disenables COPY and the cursor control keys. The keys then generate normal ASCII codes (135 to 139).

*FX5,1—selects output to the parallel (Centronics) output connector.

*FX5,2—selects the serial RS432 output. However, VDU 2 or CTRL B must still be used to start the output to the selected printer channel.

*FX5,0—selects a 'printer sink' where characters can be lost without the possibility of the system 'hanging' with a full printer buffer.

*FX6—can be used to set the character to be suppressed by the printer driver routine. For example, to prevent the 'linefeed' character (ASCII code 10) from reaching the printer *FX 6,10 is used. The *FX 6 statement should always be executed *after* the printer type has been set by *FX5.

*FX7—selects the baud rate to be used when receiving data on the RS432 interface:

*FX7,1	75 baud
*FX7,2	150 baud
*FX7,3	300 baud
*FX7,4	1200 baud
*FX7,5	2400 baud
*FX7,6	4800 baud
*FX7,7	9600 baud
*FX7,8	19200 baud

*FX8—selects transmit rate on the RS432 interface. The format is identical to the *FX7 statement.

*FX9 and *FX10—these two statements set the rate of flash of the flashing colours. For example, in MODE 2, COLOUR 9 selects flashing red-cyan.

*FX9 followed by a number (duration in hundredths of a second) will set the flash rate of the first colour—red in this case. So, to set the flash rate to one-fifth of a second:

```
*FX9,20
```

*FX10 selects the flash rate of the second colour. The format is identical to *FX9.

The default values for these statements is *FX9,50 and *FX10,50. If a value of 0 is entered for duration then that colour appears permanently on the screen.

*FX11—sets the delay before the auto-repeat starts when a key is held depressed. The duration follows the *FX11 and is measured in hundredths of a second. So, to set the auto-repeat delay to one-fifth of a second:

```
*FX11,20
```

*FX11,0—turns off the auto-repeat facility.

*FX12—sets the auto-repeat period (duration in hundredths of a second). So, to set the auto-repeat period to one second:

```
*FX12,100
```

*FX12,0—resets auto-repeat delay and auto-repeat period to normal values.

*FX15—flushes (empties) various internal buffers:

*FX15,0	all internal buffers
*FX15,1	keyboard buffer
*FX15,2	RS432 input buffer
*FX15,3	RS432 transmit buffer
*FX15,4	printer output buffer
*FX15,5	SOUND channel 0
*FX15,6	SOUND channel 1
*FX15,7	SOUND channel 2
*FX15,8	SOUND channel 3

*FX125—sets ESCAPE pressed flag. It has a similar effect to pressing the ESCAPE key.

So there is a selection of *FX statements that work on a 0.10 OS (*FX0 will tell you what you have got). Next month we will look at some of the other operating system statements including *EXEC and *SPOOL—used to MERGE programs.

contacts

Talking to other BBC microcomputer owners is a great way of learning, and can also be a great deal of fun at the same time. However, there is not always a local User Group for you to go along to in order to meet these people. The purpose of this column is to remedy that situation.

While local User Groups take some degree of organisation, talking to interested people can be as informal as you like. All the people listed in the *Contacts* column are willing to talk to, or get together with, other BBC micro owners/users on an informal basis. If you would like your name and address (or telephone number) printed in *Contacts*, please write to *Contacts*, LASERBUG, 4 Station Bridge, Woodgrange Road, Forest Gate, London E7 ONF.

We would ask you to respect the privacy of the people listed below. A letter or a telephone call is the preferred first means of contact. And please be very careful about giving addresses, and particuarly telephone numbers, to other people.

So this is the first list of BBC micro users who want contact with other BBC micro users in their area.

- John Noyce
  PO Box 450,
  Brighton, BN1 8GR
- Nick Lamb
  23 Gaywood Close,
  Caister-on-Sea,
  Great Yarmouth,
  Norfolk NR30 5RD
  0493 728442
- Dr D Yardley
  48 Park Road,
  Chandlers Ford,
  Eastleigh,
  Hants, SO5 2ES
- David Hardwick
  Hagley 885183
- Paul Firmin
  63 Lady Lane,
  Chelmsford,
  Essex CM2 OTQ
  0245 50432
- E Watson
  19 Darrell Close,
  Chelmsford,
  Essex, CM1 4EL
  0245 83790
- A Cameron
  Drummond Dores,
  Inverness IVI 2TX
  Dores 251
- Rick Branston
  37 Princes Road,
  Romford,
  Essex
  Romford 61336
- Paul Barbour
  10 Dawley Riad,
  Colnbrook,
  Slough,
  Bucks, SL3 OQH
- Chris Drage
  28 Ingersoll Road,
  Shepherds Bush,
  London, W12
- A Beer
  2 Waterworks Cottages,
  Dotton,
  Sidmouth,
  Devon, EX10 OJY

We want to encourage the setting up of local groups under the general banner of LASERBUG. All it needs is for willing volunteers to let us print their names and addresses (or telephone numbers) to allow people to contact them with a view to setting up a local group. It could be as small as a group of half-a-dozen in someone's front room, a couple of hours in a classroom after school or a larger affair at the local church hall.

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 of this project 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 coming into 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 our computer. As well as meeting new friends you can get the benefit of other people's discoveries – and share your own with them. If there is not one near you, then why not start one yourself?

We want to hear from people who want to organise a local group. Please write to us at the LASERBUG address and mark the envelope *Meeting Place.* We have already had some response from issue one of the LASERBUG newsletter. Please get in touch with the people below and let us know as soon as you get something underway. If you send us details of the User Group/Club we will print them in the newsletter on a proposed *Club Report* page.

Norman Lambert, Chairman Orpington Computer Club, 11 Vinson close, Orpington, Kent BR6 OEQ.

Dr P Voke, 39 Ligo Avenue, Stoke Mandeville, Aylesbury, Bucks, HP22 5TY. 0296 613974.

Nick Lamb, 23 Gaywood Close, Caister-on-Sea, Great Yarmouth, Norfolk NR30 5RD. 0493 728442.

Just before we sent this issue of *LASERBUG* to the printers, two books fell through our letterbox for us to review. Unfortunately, we have not had time to read through the books thoroughly. Paul, however, has given them a quick perusal to enable us to mention them here. A fuller review will follow in the next issue.

"*BASIC Programming on the BBC Microcomputer*": Neil & Pat Cryer: Prentice-Hall International: £5.95

This book is designed primarily as a beginner's guide to BBC BASIC, which is a task it performs well. The book starts off with simple statements like PRINT and INPUT and develops logically, eventually covering graphics, colour, SOUND and ENVELOPE. However, the book is not just for the beginner—there is enough in it to keep the more experienced programmer absorbed as well.

For those of you used to another dialect of BASIC, the book will provide detailed enough explanations of the statements unique to the BBC microcomputer for you to attempt 'translation' of some of your favourite programs.

All in all, at first reading stage, a book to be recommended.

"*Practical Programs for the BBC Computer and Acorn Atom*": David Johnson-Davies: Sigma Technical Press: £5.95

'*Practical Programs*' is not a book to teach you programming techniques, it is merely a collection of programs. The majority of the programs are simple and unimaginative, although the last one looks interesting—a compiler designed for SPL (a simple computer language). SPL is meant to be a limited subset of PASCAL and ALGOL.

While looking through the book, one question does spring to mind: was the book originally written for the Atom and adapted for the BBC micro at the last minute? This question arises out of the fact that none of the programs use the BBC micro to anywhere near its full capabilities.

My advice, on a cursory reading, is: read '*BASIC Programming*' thoroughly and you won't need '*Practical Programs*'.

Paul Barbour

in next issue

- ZX Spectrum vs BBC
- Computer Fair Review
- Operating System Commands—
- including Merge and Append statements
- Peek and Poke explained
- More reviews
- More programs
- More news

LIST RUN CLS CHAIN"" REN.5,5 AUTO5,5 *CAT TIME OLD

LIST RUN CLS CHAIN"" REN.5,5 AUTO5,5 *CAT TIME OLD

LIST RUN CLS CHAIN"" REN.5,5 AUTO5,5 *CAT TIME OLD