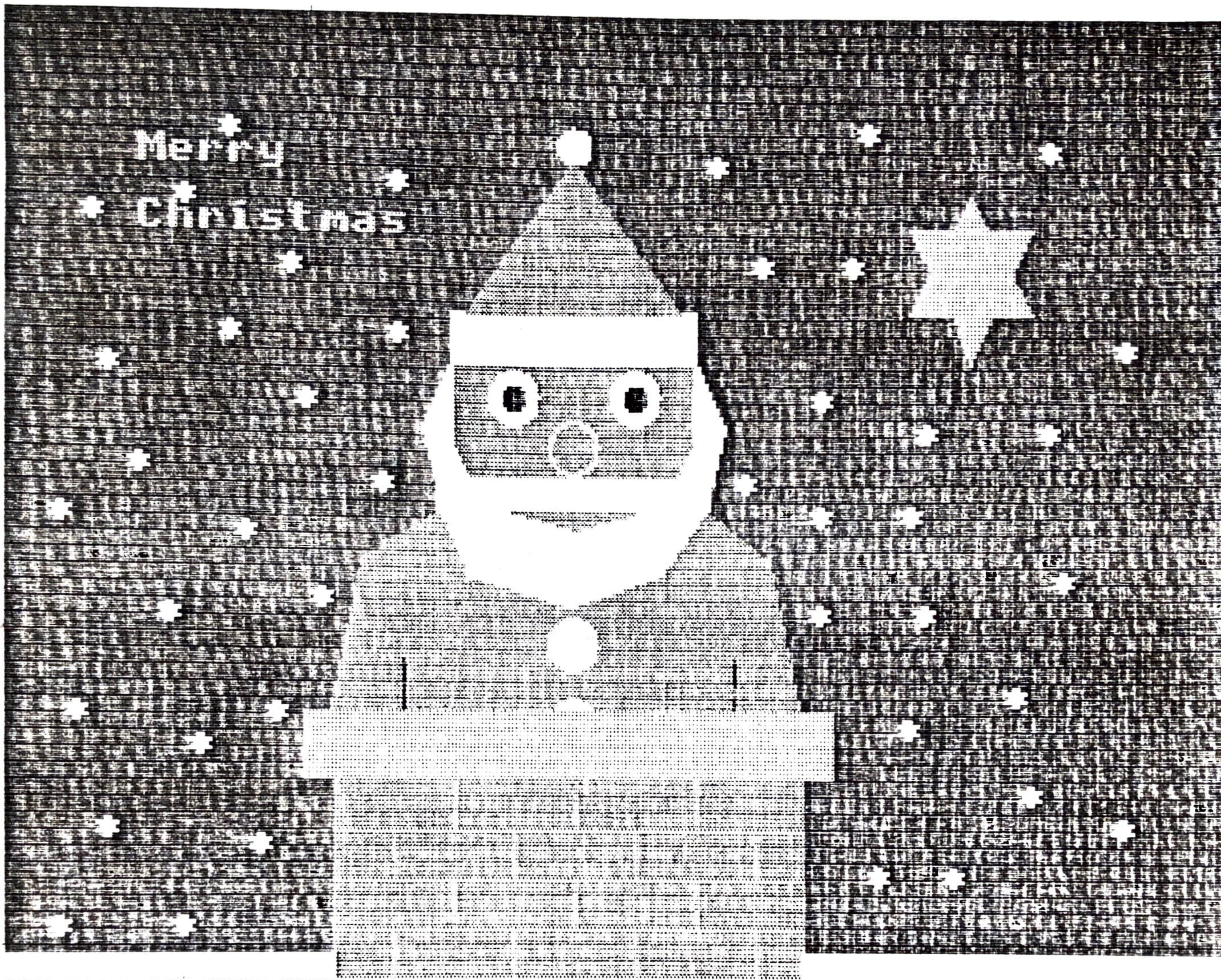
ABUG
ACORN & BBC
microcomputers
USER GROUP
Sheffield

please reply to:-



A.B.U.G. NEWSLETTER No.11
December 21st 1983

Contents

Diary

Meetings to the end of May 1984

January 4th	(Farfields)	Social evening
January 25th	(Park Baths)	BBC Buggy demonstration
February 8th	(Farfields)	Social
February 22nd	(Park Baths)	Workshop
March 7th	(Farfields)	Social
March 21st	(Park Baths)	Communications
April 4th	(Farfields)	Social
April 18th	(Park Baths)	Workshop
May 2nd	(Farfields)	Social
May 16th		Annual General Meeting

This month's newsletter consists mainly of two articles contributed by members Carol and Phil. We appreciate their contributions and reiterate our request for people to submit articles and/or programs on any subjects of interest. These should preferably be in Wordwise format on cassette or disc, but in desperation scrolls or stone tablets would be considered.

LOGO

There has recently been a resurgence of interest in the graphic language LOGO. The 'Horizon' television programme, devoted to its educational impact in primary schools, was reshown last month and many schools are now beginning to explore its potential. A full implementation of LOGO is not yet available for the BBC, although Acorn predict its arrival on chip in Spring '84 ..... Acorn Time, but LOGO-2 from Computer Concepts is still on sale. For all that this is a very elementary implementation of LOGO graphics, Computer Concepts are reporting greatly increased sales over the last few months. For an idea of the full scope of LOGO, especially the 'Turtle' graphics, there is an excellent series of articles in BYTE magazine, August 1982.

The central ideas of LOGO were introduced by Seymour Papert in his book 'Mindstorms' (Harvester Press), which is an excellent introduction to both the language itself, and to the philosophy behind it. The current role of the computer in schools is mainly as a tool for reinforcing educational concepts which have been previously introduced by the teacher. LOGO provides a situation where the child is more in command of its own learning. The child has the opportunity to communicate with the computer in simple terms, and can use the computer to solve problems. Analysis, planning, testing and modifying are all valuable activities which the child will experience at an elementary level.

LOGO can be introduced to very young children by the game of 'turtle', where one of a group of children agrees to be 'programmed' by the others, or by using a true mechanical turtle such as Bigtrack or the BBC Buggy. Bigtrack is a self-contained toy which can be programmed by the child to execute simple manoeuvres. The BBC Buggy is a much more sophisticated (and expensive) device which is connected directly to the computer. It is manufactured locally and the January meeting is devoted to a demonstration of the Buggy and its extensive software. There is a national LOGO user group (BLUG) which can be contacted through the Secretary, Pam Valley, c/o Shell Centre for Maths., School of Education, University of Nottingham.

Carol Hounslow

As a comment on the above article, the club has recently made some improvements to LOGO-2 which include a command for dumping multi-colour screens directly to an Epson printer and making the turtle directional. This has also been successfully transferred to an 8K EPROM using the program which appeared in a recent copy of Micro User, which makes it easier for young children to use. One of the better aspects of LOGO-2 is that such development is easily carried out and if anybody has any further ideas or interesting LOGO programs we would like to hear about it. As a starter, Carol has supplied the following routines which draw flowers.

```
ARC       1. REPEAT 5,FO 50:RI 18
          2. END

PETAL     1. ARC
          2. RI 90
          3. ARC
          4. END

FLOWER    1. REPEAT 12,PETAL:RI 60
          2. END
```

Wordwise Tips

Following last month's discussion of underlining problems, we contacted Computer Concepts. They confirm that there is no other solution to the underlining of a block of text which has a left margin defined. However, the underlining of centred text is possible. The technique is to always insert the centre text command (CE) first, and to separate this from the underline command (OC27,45,1 on an Epson) with a space. The text to be centred should ideally then have a compensating space at the end.
One of the constant niggles when using Wordwise for technical documents is the lack of mathematical symbols and the cumbersome manner in which most printers implement super- and sub-scripted characters. The following article discusses a program which overcomes both these problems.

User Defined Graphics on an Epson FX80 from within WORDWISE

As you are probably aware, when using Wordwise, it is possible to send specific control codes to the printer using the embedded command 'OC'. Using this method, it is quite possible to make use of the graphic facilities available on the EPSON FX80 (or any other printer) but the text becomes unwieldy, being full of embedded commands. Unfortunately, using this method the defined characters are not visible in preview mode. The control codes being sent only to the printer, making the task of producing any special text, such as equations, very difficult. I have written a program that enables up to 32 predefined characters, such as the Greek alphabet, to be used from within WORDWISE. These characters are then diplayed on the screen (in 80 column preview mode) as well as sent to the printer.

Originally I had hoped to be able to implement the use of graphics by extending the number of embedded commands, to allow an alternative character set to be switched on and off. Unfortunately this appears not to be possible so I had to resort to another method making use of the 'write character vector' to add my own routine.

Every character that is sent from the computer, to the screen or any other 'output device', passes through a special subroutine in the operating system, which we may call the 'write character routine'. To be able to modify this routine it must live in RAM rather than in ROM, which we cannot alter, and it is to this purpose that vectors exist.

The vector is a short part of the subroutine that lives in RAM and is initialised when the computer is switched on or 'break' is pressed; in fact all that the vector contains is an address pointing back inside the operating system for a jump instruction to use. What happens is this: every time something is being printed the write character subroutine jumps to the location pointed to by the 'write character vector', normally somewhere back inside the operating system and everything carries on as normal. Thus to add your own routine all that is required is to change the location where the vector points, from the operating system to the beginning of your own routine. For things to work as normal afterwards, the end of your routine must jump back to the location in ROM pointed to by the original vector.

Essentially my routine looks at every character handled by the 'write character subroutine', waiting for a special character to tell it to swap from the normal to the alternative character set or vice-versa. The correct character is then displayed on the screen and sent to the printer, if enabled. Before printing, several checks must be made, such as the current screen mode and whether the printer is enabled. Due to the way WORDISE works, in EDIT mode the control character is displayed and the user characters appear as they were typed, ie. in normal form, of course when previewing the control character is not displayed.

For a control character to switch to and from the user defined characters I chose to use the key "\", which appears as a half sign on the BBC in mode 7. Having chosen the control character it cannot now be used as normal, since it is never printed, thus it is best to choose a character that won't be required very often.

The edited text will look something like this when the alternative characters are used :-

normal text \alternative characters\ normal text

As a further embellishment my program will also print in superscripts and subscripts, the results being visible in preview mode. This is accomplished in the same manner, with a control code switching the script on and off. The whole program occupies less than 2 pages and safely lives at &D00, for a cassette filing system with the operating system 'high water mark' having being raised to &F00 (pages &9 and &A are free when using discs).

The control codes that I have chosen to use are :-

\ (half sign) switch between character sets.
^ (up arrow) switch superscript on and off.
~ (tilda) switch subscripts on and off.

I also use the '@' character as a failsafe to cancel everything at the end of the text to avoid upsetting WORDWISE.

Phil Rubini

The following text was produced using Phil's program. If anyone is interested, then see Phil directly.

αβγδεζηθικλμνξοπρστυφχψωΓΔ

abcdefghijklmnopqrstuvwxyz

πον μπορεζ κανεζ να κατασκηνωση γζα τη νυκτα any answers?

$C_3H_8 + 7O_2 = 4H_2O + 3CO_2$

$y=Ae^{\pi\theta} + \pi r^2$