Into the Arc: Peculiar Characters

by Mike Williams

In the June issue of RISC User (7:7) we published a letter about some odd hieroglyphics which a reader had uncovered in a file. We dealt briefly with this subject of alternative characters in a reply to the letter in that issue, but I thought it would be worthwhile to explore this more fully, for the benefit of those who find it all very confusing.

The Archimedes, like virtually all computers, uses a fairly standard QWERTYUIOP keyboard layout, based on the traditional typewriter style. Most readers will likely be aware that this apparently random layout originated at an early stage in the typewriter s history as a means of minimising clashes between adjacent keys. It potentially slows down the rate at which you can plug away at the keyboard; with modern computer keyboards, this consideration is quite irrelevant, but we re stuck with it.

	32	48	64	80	96	112
0	<SP>	0	@	P	‘	p
1	!	1	A	Q	a	q
2	"	2	B	R	b	r
3	#	3	C	S	c	s
4	$	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7	’	7	G	W	g	w
8	(	8	H	X	h	x
9	)	9	I	Y	i	y
10	*	:	J	Z	j	z
11	+	;	K	[	k	{
12	,	<	L	\	l	|
13	-	=	M	]	m	}

Table 1.
The ASCII codes for the keyboard characters

The typewriter analogy is useful in one other respect. Press a key on a mechanical, or electric, typewriter and your action directly causes the arm with that letter engraved on it to strike the paper through the ribbon. With a computer the processes of typing in characters, and of subsequently printing them are totally separated. Thus each letter or other character typed at the keyboard has to be stored in the computer s memory (or in a file on disc). This is accomplished by assigning to each character a code number in the range 0-255, the maximum range possible in the eight bits available. This number is called an ASCII (American Standard Code for Information Interchange) code.

The standard keyboard characters use codes in the range 32 to 127 inclusive. Space is always 32, followed by various punctuation characters. The digits 0 to 9 have codes in the range 48 to 57. The upper case alphabet runs from 65 through to 90, and the lower case alphabet from 97 through to 122. Some of the earlier RISC OS User Guides provided appropriate tables as part of the appendices, but Acorn has dropped these tables from the RISC OS 3 editions (but see Table 1). To find the ASCII code of any character add together the number at the head of its column, and the number against its row. If you wondered what has happened to the character with code 127, this doesn t print; it corresponds to the Delete key.

That leaves two gaps in the overall sequence from 0 to 255. The characters with codes from 0 to 31 inclusive are traditionally referred to as control codes . Such codes do not have a printed equivalent, but usually correspond to some kind of action. For example, the Return key generates a code of 13. So when you are typing in text, every time you press Return the code of 13 is stored in the computer s memory. Whenever the computer encounters a code of 13 in any text, it performs a carriage return function. Another example is the Tab key which generates a code of 9. However, the action performed by the computer when it encounters a code of 9 depends on the application you are running - a word processor or DTP package will often allow you, the user, to determine where tab positions are to be located.

For convenience the control keys are often represented as shown in Table 2. You will see from this table that the code of 13, which results from pressing the Return key, also corresponds to Ctrl-M. If you type Ctrl-M (hold down the Ctrl key and press the M) at any point in a text file, that also produces the effect of a Return key. You might like to see if you can work out the effect of the other control key combinations, but play safe, and make sure you have nothing visible which hasn t been already saved if it is at all important.

TOP-BIT-SET CHARACTERS

That still leaves half the available ASCII codes unaccounted for, those with codes from 128 to 255. These characters are sometimes referred to as Top-Bit-Set characters because the code for each is 128 plus some other value (and in binary 128 is the top bit).

Despite the fact that there are no corresponding keys on the keyboard, most of these codes do relate to printable characters. So how do you access them? Well there are a number of different ways of doing this. If you want to experiment, just try opening an Edit window, or open a window for your usual word processor or DTP package.

One way, if you know the ASCII code of a character, is to hold down either of the two Alt keys (on either side of the space bar), and enter the ASCII code on the numeric keypad, releasing the Alt key at the end. For example, Alt-176 entered in this way will give you a degree sign. This is all very well, but often you won t remember what the code is or have a reference handy.

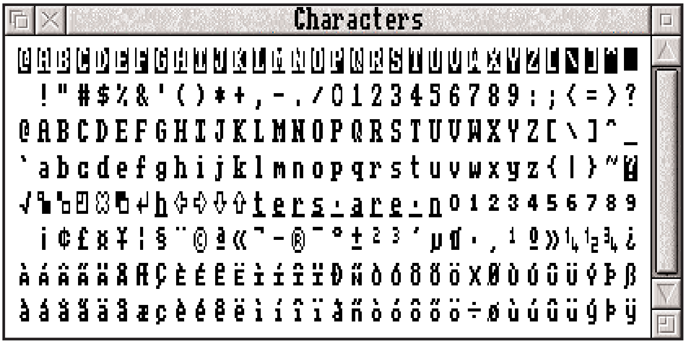
Another approach is to use the Chars application provided by Acorn. In RISC OS 3 this is one of the applications accessible from the icon bar (with RISC OS 2 it was supplied only on one of the two Apps discs). Click on the Apps icon on the icon bar, and then double-click on !Chars. This will immediately open a window on the screen showing all 256 characters in eight rows of thirty-two characters (see Figure 1). If you want to locate a corresponding ASCII code think of the columns as being numbered from 0 to 15 (left to right) and the rows numbered 0, 32, 64, 96, 128, 160, 192, 224 (top to bottom). The control codes, with ASCII codes 0 to 31, are shown in the top row as white on black.

However, Chars has further uses. Not only does

0	Ctrl-@	8	Ctrl-H	16	Ctrl-P	24	Ctrl-X	
1	Ctrl-A	9	Ctrl-I	17	Ctrl-Q	25	Ctrl-Y	
2	Ctrl-B	10	Ctrl-J	18	Ctrl-R	26	Ctrl-Z	
3	Ctrl-C	11	Ctrl-K	19	Ctrl-S	27	Ctrl-[	
4	Ctrl-D	12	Ctrl-L	20	Ctrl-T	28	Ctrl-\	
5	Ctrl-E	13	Ctrl-M	21	Ctrl-U	29	Ctrl-]	
6	Ctrl-F	14	Ctrl-N	22	Ctrl-V	30	Ctrl-^	
7	Ctrl-G	15	Ctrl-O	23	Ctrl-W	31	Ctrl-	(*)

Table 2. Traditional representation of ASCII codes 0 to 31

it show you the complete character set, but if you click over any character with the Select button, that character will be entered at the current position of the caret, and you don t have to remember any codes at all.


Characters
 ! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [\ ] ^ _
` a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~
¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬ - ® ¯ ° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿
À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
à á â ã ä å æ ç è é ê ë ì í î ï ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ

The Chars application in use

There is another feature which can be quite useful with Chars. If you click with the Menu button while the pointer is over the Chars window, a menu will appear showing a list of all the fonts currently known to your system. By default, you will see that the System Font is selected, but if you choose any other font, then the characters will be displayed in that font instead. That can be very useful with fonts like Dingbats, for example, where the characters are special symbols bearing no relationship to the normal letters of the alphabet.

Another point to note is that some outline fonts do not implement a complete character set. This is particularly true of a good few PD fonts, but may also apply for particularly fancy fonts where it is not appropriate to represent some of the more unusual characters in the chosen style.

Some individual applications may also make provision for accessing certain