

## 12. Week 12: Teletext

### 12.1 Teletext

In the early 1970s, Phillips Labs began work on transmitting digital information across the television network. The aim was to provide up-to-date news and weather information via a television set. This system was trialled first by the BBC in a system that eventually became known as “Ceefax” and then on other independent British terrestrial stations as “Oracle”. A very similar system was implemented on the BBC microcomputer, known as *Mode 7*. An example of

P100 CEEFAX 100 Mon 3 Oct 20:49/50	
<div> <div>BBC ONE</div> <div>CEEFAX</div> </div>	
NEWS HEADLINES ....101	THE DEAF ..169
NEWS IN DETAIL 102-119	SUBTITLES 170
NEWS FLASH .....150	
NEWS INDEX .....190	TV/RADIO 171-4
NEWSREEL .....199	TV CHOICE ..177
	TOMORROW ..178
FINANCE HEADLINES .120	DAY AFTER ..179
CITYNEWS .....121-139	
SPORT HEADLINES ...140	WEATHER/TRAVEL
SPORT PAGES ...130-159	INDEX .....180
FOOD GUIDE .....161	GENERAL INDEX
	.....193-5
BBC RADIO FOR SCHOOLS.....BBC2 276	

Figure 12.1: An example Ceefax page circa 1983. Taken from <http://teletext.mb21.co.uk/gallery/ceefax/main1.shtml>


	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
0	Unused/Reserved	Unused/Reserved		0	@	P	-	p
1	Red Alphanumeric	Red Graphics	!	1	A	Q	a	q
2	Green Alphanumeric	Green Graphics	"	2	B	R	b	r
3	Yellow Alphanumeric	Yellow Graphics	£	3	C	S	c	s
4	Blue Alphanumeric	Blue Graphics	\$	4	D	T	d	t
5	Magenta Alphanumeric	Magenta Graphics	%	5	E	U	e	u
6	Cyan Alphanumeric	Cyan Graphics	&	6	F	V	f	v
7	White Alphanumeric	White Graphics	'	7	G	W	g	w
8	Unused/Reserved	Unused/Reserved	(	8	H	X	h	x
9	Unused/Reserved	Contiguous Graphics	)	9	I	Y	i	y
A	Unused/Reserved	Separated Graphics	*	:	J	Z	j	z
B	Unused/Reserved	Unused/Reserved	+	;	K	←	k	1/4
C	Single Height	Black Background	,	<	L	1/2	l	
D	Double Height	New Background	-	=	M	→	m	3/4
E	Unused/Reserved	Hold Graphics	.	>	N	↑	n	÷
F	Unused/Reserved	Release Graphics	/	?	O	#	o	

Table 12.1: The control codes and characters for alphanumeric mode. Note here (because we're using white paper) foreground is shown in black and background in white. On a teletext screen we use white on a black background.

such a Ceefax screen is shown in Figure 12.1.

This project, inspired by such teletext systems, will allow a  $40 \times 25$  (1000) character file to be rendered to the screen, using similar control codes. However, some control codes are not implemented, including those to do with flashing or hidden text, and transparent backgrounds. In particular, our definition of the *double height* control code differs from that of the traditional one.

## The Control Codes

This section is based to a large extent to Richard Russell's description of Mode 7 on the BBC Micro: <http://www.bbcbasic.co.uk/tccgen/manual/tcgen2.html>.

### Coloured Text

By using the control codes 129 – 135 (0x81 – 0x87 in hexadecimal) the rest of the line will have text in the selected foreground colour.

To change the background colour, you issue a foreground colour code first, and then the "New Background" character. All the following line will now have the appropriate background colour. You'll typically then need to choose a new foreground text colour.

### Block Graphics

Teletext has a very limited ability to output low-resolution block graphics. These shapes take the place of other characters in the font and are enabled by issuing one of the coloured graphics codes e.g. *red graphics*. At this point the characters available for printing are as displayed in Table 12.2. These new graphics characters are made up of six smaller boxes, known as *sixels*. Each individual sixel has a code, either, 1, 2, 4, 8, 16 or 64 as shown in Figure 12.2. By adding these values together we can define which of these sixels are 'lit' or not. If we wish the three left-hand ones to be lit we'd use the base code (160) plus 1, 4 and 16 = 181 (0xB5 in hexadecimal). Therefore issuing the coding *green graphics* and then code 181 puts a green vertical bar on the screen.

































































	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
0	Unused/Reserved	Unused/Reserved			@	P		
1	Red Alphanumeric	Red Graphics			A	Q		
2	Green Alphanumeric	Green Graphics			B	R		
3	Yellow Alphanumeric	Yellow Graphics			C	S		
4	Blue Alphanumeric	Blue Graphics			D	T		
5	Magenta Alphanumeric	Magenta Graphics			E	U		
6	Cyan Alphanumeric	Cyan Graphics			F	V		
7	White Alphanumeric	White Graphics			G	W		
8	Unused/Reserved	Unused/Reserved			H	X		
9	Unused/Reserved	Contiguous Graphics			I	Y		
A	Unused/Reserved	Separated Graphics			J	Z		
B	Unused/Reserved	Unused/Reserved			K	←		
C	Single Height	Black Background			L	1/2		
D	Double Height	New Background			M	→		
E	Unused/Reserved	Hold Graphics			N	↑		
F	Unused/Reserved	Release Graphics			O	#		

Table 12.2: The control codes and characters for contiguous graphics mode.

	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
0	Unused/Reserved	Unused/Reserved		·	@	P	·	··
1	Red Alphanumeric	Red Graphics	·	⋮	A	Q	⋮	⋮
2	Green Alphanumeric	Green Graphics	·	·	B	R	⋮	⋮
3	Yellow Alphanumeric	Yellow Graphics	··	⋮	C	S	⋮	⋮
4	Blue Alphanumeric	Blue Graphics	·	⋮	D	T	⋮	⋮
5	Magenta Alphanumeric	Magenta Graphics	⋮	⋮	E	U	⋮	⋮
6	Cyan Alphanumeric	Cyan Graphics	·	⋮	F	V	⋮	⋮
7	White Alphanumeric	White Graphics	⋮	⋮	G	W	⋮	⋮
8	Unused/Reserved	Unused/Reserved	·	·	H	X	⋮	⋮
9	Unused/Reserved	Contiguous Graphics	⋮	⋮	I	Y	⋮	⋮
A	Unused/Reserved	Separated Graphics	⋮	⋮	J	Z	⋮	⋮
B	Unused/Reserved	Unused/Reserved	⋮	⋮	K	←	⋮	⋮
C	Single Height	Black Background	··	⋮	L	1/2	⋮	⋮
D	Double Height	New Background	⋮	⋮	M	→	⋮	⋮
E	Unused/Reserved	Hold Graphics	⋮	⋮	N	↑	⋮	⋮
F	Unused/Reserved	Release Graphics	⋮	⋮	O	#	⋮	⋮

Table 12.3: The control codes and characters for separated graphics mode.

1	2
4	8
16	<b>64</b>

Figure 12.2: Values for computing graphics codes, as added to the base code 160 (0xA0 in hexadecimal).

Notice in Table 12.2 that some other characters are still available, particularly all capital letters. This allow simple printing of capitals, even when in graphics mode, and is know as *blast-through Text*.

There is another set of block graphics, as shown in Table 12.3. For these, each sixel is separated from others by thin vertical and horizontal lines. This mode is known as *separated graphics* mode.

### Held Graphics

Generally all control codes are displayed as spaces, in the current background colour. In the held graphics mode, control code 158 (0x9E in hexadecimal), control codes are instead displayed as a copy of the most recently displayed graphics symbol. This permits a limited range of abrupt display colour changes. The held graphics character is displayed in the same contiguous/separated mode as when it was first displayed. If there has been a change in the text/graphics mode or the normal/double-height mode since the last graphics character was displayed, the held graphics character is cleared and control codes once again display as spaces. To switch held graphics mode off, use the *release graphics* control code.

### Double Height

By using the *double height* control code, characters are displayed as twice their normal size. Since they span two lines, the control codes and characters have to be repeated on the next line too, for them to be correctly displayed. The rule here, is that if a character is to be displayed as double height, the top half of the character is displayed on the first line, and the bottom half on the next line. The bottom half is only displayed as double height if the character vertically above it was also displayed in *double height* mode. The character in question need not be the same one. Note: here we deviate from other definitions of this control code.

### Some General Guidelines

- Characters are considered 7-bit (the 8th bit was typically used for parity checking over the noisy television signal). Therefore any character less than 128(0x80) should have 128 added to it. For, example if you read in character 32 (space), it should be ‘converted’ to character 160.
- Each newline on the Teletext page automatically begins with White text, single height, contiguous graphics, black background, release graphics.
- With the the exception of *hold graphics* (see above), control characters are generally rendered in the same manner as a space would be. If the background is currently red and text colour yellow, say, then the control code would show as an empty red background.

**Exercise 12.1** Implement a teletext rendering system. The 1000 character input file should be read in using `argv[1]`.

There are many ambiguities as to how various sequences of codes should be rendered. To help with this, several example files have been posted on the unit web page. If there is still doubt, make a best-guess and state your assumptions in the code. ■

(50%)

## 12.2 Testing

**Exercise 12.2** Submit the testing you have undertaken, including examples and a description of your strategies. This should convince us that you have tested every line of code, and that it works correctly. If there are still issues/bugs state them clearly. Also, point out any bugs that you have successfully found using these approaches. Submit a file named `testing.txt`, along with any other files you feel necessary in the appropriate directory.

No particular strategy is mandated. You may wish to explore a couple and briefly discuss strengths and weaknesses. ■

(25%)

## 12.3 Extensions

**Exercise 12.3** Undertake an extension of your choosing. Examples of these include:

- A system that allows you to quickly author teletext pages (perhaps using a recursive-descent parser?)
- Automatic image to teletext conversion.
- Automatic (simple) html to teletext conversion (and/or vice-versa).

Remember, that the assessment is based on the quality of your coding, so choose something to demonstrate an aspect of programming or software engineering that you haven't had a chance to use in the main assignment. Submit a file named `extension.txt` outlining, in brief, your contribution. ■

(25%)

### Hints

- Don't add graphics too early - the code is easier to test and debug with textual output to begin with.
- I advise you to use SDL for your graphics output. The library provided previously contained two functions to deal with printing characters : `Neill_SDL_ReadFont()` and `Neill_SDL_DrawString()`. The font file `m7fixed.fnt` provides the basic characters required here, but not the sixels. By understanding how the font data is rendered, the double height version of the characters should be relatively simple.
- Don't try to do all aspects of this at once - begin with coloured characters only. Add more advanced functionality later.
- Plan how you are going to store your data early in the design process. Does each character need its own data structure? Does each line? Can this be abstracted?

### Submission

Please create a directory structure, so that I can easily find the different subsections. Your testing strategy will be explained in `testing.txt`, and your extension as `extension.txt`. For the

source and extension sections, make sure there's a Makefile, so that I can easily build the code.

```
-----Top Directory-----
|           |           |
|           |           |
|           |           |
|           |           |
source      testing      extension
|           |           |
...         ... extension.txt
...         ...         ...
Makefile    Makefile    ...
              testing.txt
```

Bundle all of these up as one **single** .zip submission - not one for each subsection.