

INSTRUCTIONS

Ten Liner Teletext for the ZX Spectrum

by Matthew Begg

Introduction

Remember teletext? The simple information retrieval system broadcast to televisions around the world during the 1980s and 1990s? Well now you can relive the glory days of three-digit page numbers and blocky graphics as we bring you Ten Liner Teletext – a teletext simulator for the ZX Spectrum. It was written using only 10 lines of BASIC code, with each line containing fewer than 256 characters, and was entered in the SCHAU category of the 2022 BASIC 10Liner contest at <https://basic10liner.com>

Running the program

The tape version of Ten Liner Teletext is provided as a .tap tape image file. This can be loaded using most ZX Spectrum emulators, and has been tested working great on real Spectrum hardware using a DivMMC Future interface.

Commands

When you first run the program, you're presented with a blank page 100. Commands are displayed at the bottom of the screen:



Edit New Load Save ???

- 'Edit' to change the current page
- 'New' to create a new page
- 'Load' to overwrite the current set of pages with a previously saved set of pages
- 'Save' to save the current set of pages to tape/disk/Microdrive
- '???' to change the current page by entering a three digit page number

Editing pages

Type the command 'E' then ENTER to start editing the current page. Line numbers appear along the left-hand edge of the screen. Type the number of the line you wish to edit and press ENTER (or type '99' to exit editing mode).



Line (99 to exit)

Then start entering what you want to appear on that line. You are not limited to simple white text on a black background – you can use the GRAPHICS and INK/PAPER colours on the top row of the Spectrum keyboard to make your pages come alive with vibrant colours and quadrant graphics.



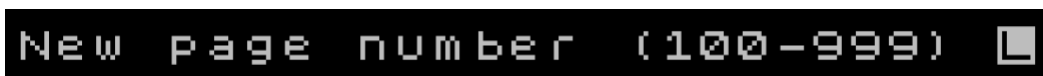
For example:

- CAPS SHIFT + SYMBOL SHIFT (to enter EXTEND MODE) then '1' will give you a paper colour of BLUE
- EXTEND MODE then CAPS SHIFT + '2' will give an ink colour of red
- CAPS SHIFT + '9' will enter GRAPHICS mode. Keys 1 to 8 will then give you the different graphics characters displayed on the keys (CAPS SHIFT and the number gives you the inverse character)
- EXTEND MODE then '9' enters BRIGHT mode (giving different shades of the 7 colours for a total of 15 colours)
- EXTEND MODE then CAPS SHIFT + '9' enters flashing mode

Use combinations of the above techniques to produce colourful teletext pages, just like the ones back in the day.

Creating a new page

Type the command 'N' then ENTER to create an additional teletext page. You'll be asked for a three digit number between 100 and 999 for the new page.



You'll then be taken to that new empty page. Use the Edit command (see above) to start filling in the blank page.

Saving and loading sets of pages

Once you have created your set of pages, you'll want to save them to use later (otherwise they are lost when the Spectrum is turned off). There are four versions of the program that allow saving to different types of media. You can see which version you are running by looking at the yellow code at the top right:

TAPE – the most common version allows you to save your pages to cassette tape. Works on ZX Spectrum 16k*/48k/128k/+2.

DISK – uses 3 inch floppy disks. Works on ZX Spectrum +3 only.

MD – uses ZX Microdrive cartridges. Works on ZX Spectrum 16k*/48k/128k using the Interface 1.

NET – allows one Spectrum to act as a broadcaster of pages with up to 63 other Spectrums receiving the pages. Works on ZX Spectrum 16k*/48k/128k using the Interface 1 and appropriate 3.5mm jack cables.

Type the command 'S' then ENTER to save your set of pages to tape/disk/Microdrive.

Type the command 'L' then ENTER to overwrite your current set of pages with a set from tape/disk/Microdrive.

APPENDIX 1 - Broadcasting pages via the ZX Interface 1 network

On the NET version, the Save command has been replaced by the Broadcast command. To use this feature, type the command 'L' then ENTER on all of the receiving ZX Spectrums first, then type the command 'B' then ENTER on your transmitting ZX Spectrum containing a set of pages. The set of pages will then be sent to all ZX Spectrums on the network at the same time.

APPENDIX 2 - Compatibility

As it comes, the Ten Liner Teletext system can store up to 20 teletext pages. This can comfortably fit in the memory of the 48k and 128k Spectrums. However, you can alter the program to store more or fewer pages. To do this, type the command 'E' then ENTER, followed by SYMBOL SHIFT + A (for the 'STOP' command) then ENTER. This will stop the program, and allow you to alter the program listing. Edit line 10 of the program (CAPS SHIFT + 1 for EDIT) and change the max=20 to max=nn where nn is between 2 and 35. Some suggestions

- * For 16k ZX Spectrums, you can alter this to a maximum of 6 otherwise you'll get an 'Out of memory' error
- For 48k and 128k ZX Spectrums, you can go up to 35 comfortably. The default is 20 as a compromise of loading/saving times on tape, but clearly if you're using disk or Microdrive, this is less of a concern.
- Currently there is no benefit to running the system on Spectrums with 128k of memory as it cannot access the additional memory. This is potentially possible, and I invite people to modify the program to access more than 30 pages (without increasing the program size beyond 10 lines though of course!)

APPENDIX 3 - BASIC listing (TAPE version)

```
10 CLEAR : LET max=20: BORDER 0: PAPER 0: INK 7: DIM p$(max,20,53): LET p$(1,1)="100":  
LET p=1: LET f=2: LET i=1: REM Ten Liner Teletext by Matthew Begg © 2022. Set max between 6  
(16k) and 35 (48k)
```

```
20 CLS : PRINT "P";p$(p,1,1 TO 3);" Ten Liner Teletext TAPE ": FOR n=1 TO 20: PRINT AT  
n,0;p$(p,n,4 TO 53);: NEXT n: IF i=1 THEN INPUT "Edit New Load Save ??? "; LINE c$: REM ***  
Displays the current page
```

```
30 IF c$>="100" AND c$<="999" THEN GO TO 90: REM *** Change the current page
```

```
40 IF c$="L" OR c$="I" THEN INK 0: LOAD "Pages" DATA p$(): INK 7: LET p=1: LET f=VAL  
p$(1,2,1 TO 3): LET max=VAL p$(1,3,1 TO 3): REM *** Load pages
```

```
50 IF c$="S" OR c$="s" THEN LET p$(1,2,1 TO 3)=STR$ f: LET p$(1,3,1 TO 3)=STR$ max: INK  
0: SAVE "Pages" DATA p$(): INK 7: REM *** Save pages
```

```
60 IF (c$="N" OR c$="n") AND f<max+1 THEN INPUT "New page number (100-999) "; LINE  
p$(f,1,1 TO 3): LET p=f: LET f=f+1: REM *** Creates a new page
```

```

70 IF c$="E" OR c$="e" THEN FOR n=1 TO 20: PRINT AT n,0;"";n;"";: NEXT n: INPUT "Line (99
to exit) ";n: LET i=1: IF n<=20 THEN LET i=0: PRINT AT n,0;"";n;"";: INPUT (n);": LINE I$: LET
p$(p,n,4 TO 4+LEN I$)=I$: REM *** Edit a line

80 GO TO 20: REM *** End of main loop

90 FOR n=1 TO max: IF p$(n,1,1 TO 3)=c$ THEN LET p=n: GO TO 20: REM *** Search for
required page

100 NEXT n: GO TO 20: REM *** Search continues

```

APPENDIX 4 – Description of how the code works

Variables

max – integer storing the maximum number of pages the system can store (see Appendix 2 for pros and cons of changing this value)

p\$ - array storing 'max' number of pages, each containing 20 lines of text that can be up to 53 characters long. Why 53 when the Spectrum display is only 32 characters wide? Well, the first 3 characters of the first line are used to store the three digit page number. The first 2 characters of the second line are used to store the next page index ('f'). The first 2 characters of the third line are used to store the maximum number of pages allowed ('max'). Also the Spectrum graphics commands are stored as hidden characters and use up 1 character, so there are 18 additional characters allowed for changing ink/paper/bright/inverse properties

p – integer storing the current page index. Default is 1 (where the blank page 100 is stored)

f – integer storing the next page index. Default is 2

i – integer used as a flag for whether to prompt for a command. Used by the edit command to keep the user in the edit routine unless they type '99' to exit

n – integer used temporarily for FOR...NEXT loops

c\$ - string used to store the command typed by the user

Line by line description

10 – initialises the system. Clears all variables. Sets the maximum number of pages (max). Sets the screen and border to black, with text displayed in white. Creates the array that stores the set of pages (p\$). The first blank page is created and given the page number "100".

20 – beginning of the main loop that displays the current page. The screen is cleared, the header row is displayed (showing the current page number and which media method is in use, e.g. 'TAPE'). A loop then displays all 20 lines of the current page. If not in page editing mode (i.e. i=1) then prompt for a page number or command to be entered.

30 – change the current page. If a number between 100 and 999 is entered then jump to line 90 which starts the search for that page

40 – loading a set of pages. The entire set of pages is stored as the p\$ array. Sinclair BASIC doesn't easily allow more than one variable to be saved in one go, which is why the program uses a single array p\$ to store everything needed to reproduce a set of pages. Once loaded, it jumps to the first page (p=1) and sets the maximum number of pages (max)

50 – saving a set of pages.

60 – creates a new teletext page (assuming the maximum number of pages hasn't been reached). Asks the user for a three digit page number, then creates the page.

70 – enters page editing mode. Writes the line numbers along the left edge of the screen, then asks the user which line they would like to edit. This is the only point in the program where the user can potentially stop the program and alter the listing (by entering the STOP command). It then asks the user to enter text for that line, then appends it into the main array (p\$). The 'i' flag is used to keep users in the editing mode unless '99' is entered

80 – end of the main loop. Goes back to line 20 to display the current page again

90 – comes from line 20 to search for a page based on the three digit page number. Loops through the main array looking for that page number. If found, it sets the current page index accordingly and jumps to line 20 to display the page.

100 – ends the loop searching for the page, then jumps to line 20 even if the page isn't found (leaving you on the current page)

Matthew Begg
Manchester, UK
February 2022