

## GFXterm - VT/ANSI Terminal Emulation with Graphics Support

### Overview:

GFXterm has been designed as a simple terminal emulator for use with Geoff Graham's single-chip Micromite computers that run MMbasic. As such it provides just enough VT100/220 emulation to use the Micromite's inbuilt editor with an 80 column by 24 line screen size.

In addition, GFXterm supports a simple set of graphics extensions that are suitable for drawing very basic graphs. Lines and arcs can be drawn, enclosed regions filled, and rectangular blocks scrolled in any direction. Graphics are drawn on a separate 'glass layer' overlaying the normal text screen. Text and graphics layers operate completely independently and do not in any way interact, with the text layer visible through blank areas of the graphic layer (wherever the graphic pixels are black: R=0, G=0, B=0).

### Operation:

Run the program "GFXterm.exe". The only thing needed to run GFXterm is this single .exe file. Upon start-up you will be presented with a black screen, with the terminal in the disconnected state. To connect to an attached Micromite, right-click on the terminal screen, and select **CONNECT**. When prompted type in the comm port name that the Micromite is attached to/as, followed by a colon, followed by the console baud rate (Micromites default to 38400 baud). Press **ENTER** and you should be connected - if there is something wrong an error window will pop up.

The right-click menu is the main method of controlling GFXterm. The menu items available are:

**CONNECT / DISCONNECT** - used to connect to or disconnect from a Micromite. Normally GFXterm will remember the last successfully used connection string. If your Micromite is attached to COM3 and you have not changed the console baud rate, then the connection string will be:

com3:38400

**LOG to file / STOP logging** - used to save all output from the Micromite to a text file. Only the text is saved, no colour or location information, so logging to a file while the Micromite's inbuilt editor is in use will produce a great deal of garbage in the log file. Normally logging will be used to save program output, or in conjunction with `LIST ALL` to save a program held within the Micromite to your PC.

**paste (from clipboard or from text file)** - these two options can be used to upload a program to the Micromite. GFXterm detects if pasting into the Micromite's inbuilt editor and slows down to accommodate. If at the command prompt, you can first type `AUTOSAVE` to quickly save a program to the Micromite's flash memory.

HINT: if pasting into the inbuilt editor, ensure there is a space character to the right of the cursor before pasting to suppress automatic indenting.

**CANCEL paste** - immediately cancel any paste operation that is in progress, for use in cases of inadvertently pasting from a long file. This function is also mapped to the **alt-Z** key combination.

**font size (9/12/14 point)** - selects a larger screen font. The default is 9 point, but if changed GFXterm remembers the setting. Note that changing the font size also changes the horizontal and vertical pixel counts for graphics, but that these values can be read from GFXterm by the Micromite.

**font colour (Red, Green, Yellow, Blue, Magenta, Cyan, WHITE)** - this is the default text colour, and is remembered by GFXterm. The default setting is **WHITE**. Note that the Micromite can over-ride this setting, as happens when the inbuilt editor is used with `OPTION COLOURCODE ON`.

**clear (GFX plane, text plane, ring buffer)** - these are also mapped to **alt-C**, **alt-D**, and **alt-A** respectively. The most useful one of these will be **alt-C** where a program has drawn on the graphic layer and you want to clear this so as to see underlying text when in the Micromite's inbuilt editor.

**select and copy** - this displays a monochrome text-only view of the screen from which it is possible to select text with the mouse and copy it to the clipboard using **ctrl-C**. The view is of a frozen snapshot, and the remainder of GFXterm continues to operate in the background. Press **ENTER** or **ESC** to exit this view.

**command window** - this pops up a small window that displays all VT/ANSI and GFX commands that have been precessed. This can be exceptionally useful when trying to figure out what has gone wrong when using graphics, as well as obtaining a better understanding on how the Micromite's inbuilt editor works. The command window can be closed with the **X** in its top right corner.

### Keyboard shortcuts:

**alt-Z** cancel any paste operation that is in progress

**alt-A** clear ring buffer. This is a 64k buffer used for serial input.

**alt-C** clear graphics layer

**alt-D** clear text layer

**alt-B** send 'break' signal, this instructs 1455 firmware to reset the Micromite

**alt-M** insert a dividing line in the command window output

### VT/ANSI commands:

See the document "VT220 partial escape sequence list.pdf" for a list of the commands that have been implemented.

Colour for text and background is supported, with 8 possible colours for each. Background colours are dimmed, while foreground colours are bright by default. However, a dim foreground attribute (2) is available to effectively give 16 foreground colours. The blinking attribute (5) is ignored.

Scroll regions can be set with **DECSTBM**, but only **IND**, **NEL**, and **RI** make use of these. Commands for deleting and inserting characters are currently not supported, as the Micromite's inbuilt editor does not make use of these. Setting and clearing tab stops is also not supported, with tab stops instead fixed at every 8 characters.

## GFX commands:

The syntax of a GFX command string is as follows:

**<DLE> command parameter,..., parameter <CR> [<LF>]**

where the command and parameters can be separated by spaces, commas, semicolons, or tab characters. The command string is terminated by a carriage return, with any following line feed ignored. **<DLE>** (data link escape) is `chr$(16)`.

Commands are detailed in the file "GFX commands.pdf". The best understanding of how to use GFX commands is to look at the sample programs provided, "GFX demo 2.bas" and "GFX bouncing ball.bas". The commands are, on the whole, just wrappers for windows graphic object commands.

Note that the horizontal and vertical pixel counts are dependant on the font size selected in GFXterm, hence any program using graphics should first issue the "?" command to retrieve these counts. the "ink" command accepts red, green, and blue parameters each ranging from 0 to 255. An ink colour of 0,0,0 is transparent - for black use 1,1,1 instead.

The "arc" command can draw circles, ellipses, or parts thereof. The angles specified are in degrees, with 0 degrees due north (12 o'clock), positive values moving clockwise. The "fill" command expects a fully enclosed area bounded by the current ink colour, and flood fills the enclosed region with that colour. You can not fill with a different colour.

To prevent filling the ring buffer, a program can synchronize with GFXterm using the control codes **<ENQ>** and **<ACK>** (`chr$(5)` and `chr$(6)` respectively). When GFXterm sees an **<ENQ>** as it processes serial data it responds by sending an **<ACK>**.

The keyboard shortcut **alt-C** clears the graphics layer - when creating scrolling graphs this can be extremely useful.

Robert Rozee  
8-September-2017