GFXterm - VT100/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 running MMbasic. As such, it provides just enough VT100/ANSI 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 rolling graphs. Lines and arcs can be drawn, enclosed regions filled, and rectangular areas scrolled in any direction. Graphics are drawn on a separate 'glass layer' overlaying the normal text screen. This layer is turned off be default, being turned on by any successful graphics command being processed. The graphics layer can then be turned off again by pressing **alt-C** to clear graphics. GFXterm runs slightly faster with graphics turned off.

Text and graphics layers operate completely independently and do not in any way interact, with the text layer visible through 'clear' areas of the graphics layer (wherever pixels are set to the colour: R=0, G=0, B=0).

Operation:

Run the program "GFXterm.exe" (Win32), GFXterm64 (64-bit Linux), or GFXterm32 (32-bit Linux). The only thing needed to run GFXterm is this single executable file. Upon start-up you will be presented with an empty terminal window and flashing red block cursor, with the terminal sitting in the disconnected (local loopback test) state - ascii keypresses will be printed to the screen and control characters processed.

To connect to a local or remote Micromite, right-click anywhere on the terminal window (or press **alt-M**), and select **CONNECT** from the pop-up menu that appears. You will be prompted to enter a comm port and baud rate (note that by default the Micromite console runs at 38400 baud),. Just make the required selection, press **ENTER**, and you should be connected to the Micromite.

The right-click menu is the main method of controlling GFXterm, although a number of the functions available in the menu are also mapped to shortcut (**alt-**) keys. There are also a few functions that are only accessible through shortcut keys.

Menu Commands:

CONNECT / DISCONNECT - used to connect to or disconnect from a Micromite. The comm port and baud rate are prompted for, seperated by a colon. Other serial port parameters are hard-wired to 8 data bits, 1 stop bit, no parity, no flow control. ~~If the 'network' tab is clicked, a remote (network) host name and port number (separated by a colon) can instead be entered. Any port can be used - the connection is raw 8-bit ASCII data, with no specific protocol or any sort of data encryption employed.~~

Normally GFXterm will remember the last successfully made serial ~~or network~~ connection, this connection may be restored directly with the **ctrl-SPACE** keyboard shortcut.

LOG to file / STOP logging - used to save terminal output from the Micromite to a text file. Only plain text is saved, no colour or cursor location information. 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. The STOP logging function is also mapped to the **alt-L** key combination, a second press will resume logging - appending to the existing log file.

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

paste (from clipboard / 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. While at the MMbasic command prompt, you can first type **AUTOSAVE** to quickly save a program directly to the Micromite's flash memory. Pasting from the clipboard is also mapped to the **alt-P** key combination.

HINT: if pasting into the Micromite's inbuilt editor, always ensure there is a space character to the right of the cursor before pasting in order to suppress automatic line indenting.

CANCEL paste - immediately cancels any paste operation that is in progress, in case of inadvertently pasting from an unintended source. This function is also mapped to the **alt-Z** key combination.

---

screen font - select from available fonts and sizes. The default size is 12 point "Monospace", and if changed GFXterm remembers the new setting (along with font colour, dimmable text, and diagnostics options below). Note that changing the font family and size also changes the horizontal and vertical pixel counts for graphics, but that these values can be read from GFXterm by the Micromite before using any graphics commands.

font colour (Red, Green, Yellow, Blue, Magenta, Cyan, WHITE, swap B/W) - selects the default text colour. The default colour setting is WHITE , but the Micromite can always override this setting, as happens when the inbuilt editor is used with `OPTION COLOURCODE ON`. swap B/W is black text on a white background.

dimmable text (enabled, bright #1, bright #2) - selects how the **SGR 2** (dim foreground) escape sequence is handled. When enabled (default), the foreground colour can be selected between bright and dim with **SGR 2** while the background colour is always dim. Selecting bright #1 overrides **SGR 2**, forcing foreground colours to always be bright, while non-black background colours are also forced to bright when bright #2 is selected.
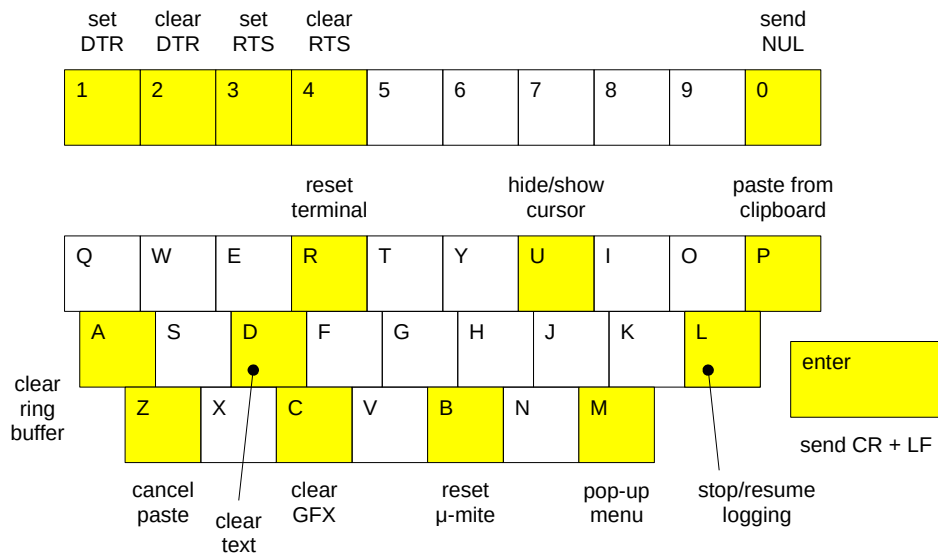
---

clear/reset (GFX layer, text layer, ring buffer, RESET terminal) - these commands are also mapped to the keys **alt-C**, **alt-D**, **alt-A** and **alt-R** respectively. The most useful one of these will be **alt-C** where a program has drawn on the graphics layer and you want to clear this so as to see underlying text when going back into the Micromite's inbuilt editor.

diagnostics - these options are intended for use when GFXterm is started from a Linux Terminal Session. When started in this manner, GFXterm can be set to continuously print information about:
- Rx packet timing, along with a plain-text version of the data with decoded hex values for control characters and the top 128 non-ASCII codes, or,
- a running list of VT and GFX commands processed, along with their pass/fail status.

Keyboard Shortcuts:

alt-Z　　　　CANCEL any paste operation that is in progress　　*(also a menu item)*
alt-A　　　　clear ring buffer (64k buffer used for serial input)　　*(also a menu item)*
alt-C　　　　clear graphics layer　　*(also a menu item)*
alt-D　　　　clear text layer　　*(also a menu item)*
alt-P　　　　paste from clipboard　　*(also a menu item)*
alt-L　　　　STOP/RESUME logging　　*(also a menu item)*
alt-R　　　　RESET terminal　　*(also a menu item)*
alt-> and <　bell volume level up and down
alt-U　　　　hide/show cursor
alt-B　　　　send break, signals 1455 firmware to reset Micromite
alt-ZERO　　send null character
alt-ENTER　send CR + LF character sequence
alt-1 to 4　set/clear modem control (DTR/RTS) outputs
ctrl-SPACE　try to restore last valid connection, or disconnect if already connected

Command-line Switches:

GFXterm supports the following command-line switches to help automate connection:

GFXterm /s commport[:baudrate]

The baudrate is optional, defaulting to 38400 if not specified. If the commport can not be found, GFXterm will start up normally in the offline state.

GFXterm /n hostname:port

Connect to a network host. Both hostname and port must be specified.

GFXterm /e

Enable extended format logging. Every received packet will be timestamped, and control characters replaced by a human-readable form (<hh> or <label>). This switch can not be combined with any others. Log files may become very large with extended logging.

<u>Status Line</u>:

Above the terminal screen area is a status line that displays various useful pieces of information about the terminal session. This includes last ASCII character displayed and key pressed, current cursor location, timing information, and various status indicators:

```
00  row=01  col=01  key=00  000mS  000mS   00%   online   logging   [000000]   (00,00)
1.    2.      3.      4.      5.     6.      7.      8.         9.       10.        11.
```

1.	last ASCII character displayed	
2.	cursor position, current row	
3.	cursor position, current column	
4.	last ASCII key pressed	
5.	time since last serial data received	
6.	time since last serial data transmitted	
7.	percentage of ring buffer in use	
	silver	<10% used
	blue	10% to 39% used
	yellow	40% to 69% used
	red	>70% used
8.	online annunciator	
	green	local serial port
	yellow	TCP/IP connection
9.	file logging annunciator (yellow when logging to file)	
10.	number of characters waiting to be pasted:	
	green	fast paste
	blue	slow paste
11.	character cell location of mouse within the terminal screen area	

<u>VT100/ANSI Commands</u>:

See the document "VT220 partial escape sequence list.pdf" for a full list of the commands that have been implemented (marked with ticks). The list mostly covers the basic VT100 and VT102 commands, with a few VT220 specific additions and ANSI colour support. Note that VT52 compatibility mode has not been implemented.

Colour for text and background (using **SGR**) is supported, with 8 possible colours for each. Background colours are dimmed, while foreground colours are bright by default. However, a dim foreground attribute (**SGR 2**) is available to effectively give 16 foreground colours if dimmable text has not been disabled. The blinking attribute (**SGR 5**) is ignored.

The scroll region can be set with **DECSTBM**, but only **IND**, **NEL**, **RI**, **CUU**, **CUD**, **IL**, **DL**, and **LF** make use of the top/bottom margins set. Character delete/insert/erase is supported with **DCH, ICH** and **ECH,** while character insert/replace mode is selectable via **RM / SM 4**. Setting and clearing tab stops is not supported, with tab stops instead fixed at every 8 columns.

AutoWrap at end-of-line is hard-wired on, and cursor positioning is permanently fixed to absolute - location (1,1) being top-left of the terminal screen irrespective of any top or bottom margins set with **DECSTBM**.

Mouse scroll wheel activity is mapped to the cursor up/down keys (unshifted) and cursor left/right keys (shifted), and will work with the Micromite's internal editor. Further to this, mouse position reporting (X10, VT200, and extensions thereof) can be turned on/off with

SM / RM ?9, **?1000**, **?1006**, and **?1015**, thereby providing simple mouse support via single-click (X10) and button down/up (VT200) pointer location only - no drag-and-drop, mouse tracking, etc. Modifiers of SHIFT, CTRL, and ALT are detectable in the VT200 mouse mode.

The text cursor can be hidden/shown using **RM / SM ?25** - this may also be manually controlled from the keyboard using **alt-U** to toggle the cursor state.

GFX Commands:

The syntax of a GFX command string is as follows:

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

where the command and parameters (all in plain text) can be separated by spaces, commas, semicolons, or tab characters. A GFX command string is terminated by a carriage return, with any immediately following line feed skipped. **<DLE>** (data link escape) is **chr$(16)**.

For example, to draw a circle with centre at (100,100), radius 50, coloured green using a brush 3 pixels wide, the following lines of MMbasic code would be used:
```
PRINT chr$(16) "i" 0, 255, 0, 3
PRINT chr$(16) "a" 50, 50, 150, 150, 0, 0
```

Commands are detailed in the file "GFX commands.pdf". They can be written in full, or abbreviated to the first letter, and can be upper or lower case. The best way to understand how to use GFX commands is to look at the two sample programs provided: "GFX demo 2.bas" and "GFX bouncing ball.bas". The commands are, on the whole, just wrappers for windows (win32) graphic object commands.

The following points are worth noting:

1. 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 and scale output accordingly.

2. The origin for all graphics commands is the bottom left corner of the terminal window.

3. The "**ink**" command accepts red, green, and blue parameters with each ranging from 0 to 255. An ink colour of 0,0,0 is transparent - for opaque black use 1,1,1 instead.

4. 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 - set start and finish as both 0 for a full circle or ellipse.

5. The "**fill**" command expects a fully enclosed area bounded by the current ink colour, flood filling the enclosed region with that same colour. You can not fill with a different colour.

To help prevent overflowing the serial input 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 incoming serial data, it responds by sending an **<ACK>** in reply. This allows a running MMbasic program to wait for GFXterm to catch up.

<u>Other Notes</u>:

GFXterm has an internal 64k ring buffer that holds incoming serial data before it is displayed on-screen. In normal use only a very small portion of this ring buffer is used, however under certain extreme circumstances it is possible to fill the buffer. For instance, an MMbasic program that sits in a tight loop rapidly outputting data for an extended period of time, combined with manually selecting a non-bitmapped screen font (non-bitmapped fonts take considerably more time to display on the screen).

When the ring buffer becomes more than 98% full, GFXterm responds by suspending the printing of characters to the screen - the cursor position still updates, and the screen still scrolls as normal, but no characters appear. When the ring buffer drops below 95% full, character printing resumes. In most cases this should prevent the ring buffer ever reaching 100% full (at which point incoming serial data is simply discarded).

Remember: the keyboard shortcut **alt-C** *clears the graphics layer - when creating scrolling graphs this can be extremely useful to allow viewing the text underneath.*

Robert Rozee
20-May-2020