WIKIPEDIA

VGA text mode

VGA text mode was introduced in 1987 by IBM as part of the VGA standard for its IBM PS/2 computers.[1] Its use on IBM PC compatibles was widespread through the 1990s and persists today for some applications on modern computers.[2] The main features of VGA text mode are colored (programmable 16 color palette) characters and their background, blinking, various shapes of the cursor (block/underline/hidden static/blinking),[3] and loadable fonts (with various glyph sizes).[4] The Linux console traditionally uses hardware VGA text modes,[5] and the Win32 console environment has an ability to switch the screen to text mode for some text window sizes.

Contents

Distinctive features of VGA text as it commonly used:

Light gray background (normally not white).

Box-drawing.

Various back-/foreground combinations.

CGA–EGA-style 16 color palette for foreground. Blinking text.

Cursor.

Data arrangement

Text buffer

Each screen character is represented by two bytes aligned as a 16-bit word accessible by the CPU in a single operation. The lower, or character, byte is the actual code point for the current character set, and the higher, or attribute, byte is a bit field used to select various video attributes such as color, blinking, character set, and so forth.[6] This byte-pair scheme is among the features that the VGA inherited from the EGA, CGA, and ultimately from the MDA.

Attribute								Character							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Blink[n 1]	Background color			Foreground color[n. 2][n. 3]				Code point							

1. ^ Depending on the mode setup, attribute bit 7 may be either the blink bit or the fourth background color bit (which allows all 16 colors to be used as background colours).

2. ^ Attribute bit 3 (foreground intensity) also selects between fonts A and B (see below). Therefore, if these fonts are not the same, this bit is simultaneously an additional code point bit.
3. ^ Attribute bit 0 also enables underline, if certain other attribute bits are set to zero (see below).

Colors are assigned in the same way as in 4-bit indexed color graphic modes (see VGA color palette). VGA modes have no need for the MDA's reverse and bright attributes because foreground and background colors can be set explicitly.

Underline

The VGA hardware has the ability to enable an underline on any character that has attribute bit 0 set. However, since this is an MDA-compatible feature,[7] the attribute bits not used by the MDA must be set to zero or the underline will not be shown.[6] This means that only bits 3 (intensity) and 7 (blink) can be set concurrently with bit 0 (underline).[8] With the default VGA palette, setting bit 0 to enable underline will also change the text colour to blue. This means text in only two colors can be underlined (5555FF and 0000AA with the default palette).

Despite all this, the underline is not normally visible in color modes, as the location of the underline defaults to a scanline below the character glyph, rendering it invisible.[6] If the underline location is set to a visible scanline (as it is by default when switching to an MDA-compatible monochrome text mode), then the underline will appear.

Fonts

Screen fonts used in EGA and VGA are monospace raster fonts containing 256 glyphs. All glyphs in a font are the same size, but this size can be changed. Typically, glyphs are 8 dots wide and 8–16 dots high, however the height can be any value up to a maximum of 32. Each row of a glyph is coded in an 8-bit byte, with high bits to the left of the glyph and low bits to the right. Along with several hardware-dependent fonts stored in the adapter's ROM, the text mode offers 8[6] loadable fonts. Two active font pointers (font A and font B) select two of the available fonts, although they usually point to the same font. When they each point to different fonts, attribute bit 3 (see above) acts as a font selection bit instead of as a foreground color bit. On real VGA hardware, this overrides the bit's use for color selection, but on many clones and emulators, the color selection remains — meaning one font is displayed as normal intensity, and the other as high-intensity. This error can be overcome by changing the palette registers to contain two copies of an 8-color palette.

Norton Utilities 6.01, an example of advanced TUI which redefines the character set to show tiny graphical widgets, icons and an arrow pointer in text mode.

There are modes with a character box width of 9 dots (e.g. the default 80×25 mode), however the 9th column is used for spacing between characters, so the content cannot be changed. It is always blank, and drawn with the current background colour.[6] An exception to this is in *Line Graphics Enable* mode, which causes code points 0xC0 to 0xDF inclusive[6] to have the 8th column repeated as the 9th. These code points cover those box-drawing characters which must extend all the way to the right side of the glyph box. For this reason, placing letter-like characters in code points 0xC0–0xDF should be avoided. The box-drawing characters from 0xB0 to 0xBF are not extended, as they do not point to the right and so do not require extending.

Cursor

VGA shows us the code page 737 with Greek letters

The shape of the cursor is restricted to a rectangle the full width of the character box, and filled with the foreground color of the character at the cursor's current location. Its height and position may be set to anywhere within a character box;.[9] The EGA and many VGA clones allowed a split-box cursor (appearing as two rectangles, one at the top of the character box and one at the bottom), by setting the end of the cursor before the start, however if this is done on the original VGA, the cursor is completely hidden instead.[9] The VGA standard does not provide a way to alter the blink rate,[9] although common workarounds involve hiding the cursor and using a normal character glyph to provide a so-called software cursor.

Mouse cursor in Impulse Tracker

A mouse cursor in TUI (when implemented) is not usually the same thing as a hardware cursor, but a moving rectangle with altered background or a special glyph.

Some text-based interfaces, such as that of Impulse Tracker, went to even greater lengths to provide a smoother and more graphic-looking mouse cursor. This was done by constantly re-generating character glyphs in real-time according to the cursor's on-screen position.[10][11][12][13]

Access methods

There are generally two ways to access VGA text-mode for an application: through the Video BIOS interface or by directly accessing video RAM[4] and I/O ports. The latter method is considerably faster, and allows quick reading of the text buffer, for which reason it is preferred for advanced TUI programs.

The VGA text buffer is located at physical memory address 0xB8000.[14] Since this address is usually used by 16-bit x86 processes operating in real-mode, it is also the first half of memory segment 0xB800. The text buffer data can be read and written, and bitwise operations can be applied. A part of text buffer memory above the scope of the current mode is accessible, but is not shown.

The same physical addresses are used in protected mode. Applications may either have this part of memory mapped to their address space or access it via the operating system. When an application (on a modern multitasking OS) does not have control over the console, it accesses a part of system RAM instead of the actual text buffer.

For computers in the 1980s, very fast manipulation of the text buffer, with the hardware generating the individual pixels as fast as they could be displayed, was extremely useful for a fast UI. Even on relatively modern hardware, the overhead of text mode emulation via hardware APA (graphics) modes (in which the program generates individual pixels and stores them into the video buffer) may be noticeable.

Modes and timings

Video signal

From the monitor's side, there is no difference in input signal in a text mode and an All Points Addressable (APA) mode of the same size. A text mode signal may have the same timings as VESA standard modes. The same registers are used on adapter's side to set up these parameters in a text mode as in APA modes. The text mode output signal is essentially the same as in graphic modes, but its source is a text buffer and character generator, not a framebuffer as in APA.

PC common text modes

Depending on the graphics adapter used, a variety of text modes are available on IBM PC compatible computers. They are listed on the table below:

Mode(s) (decimal)	Mode(s) (hex)	Type	Text res. (W×H)	Char. size	Graphics res.	Colors / memory model	Adapters
7	0007h	VGA Text	80×25	9×14	720×350	2 (mono) / MTEXT	MDA, Hercules[15]
6	0006h	VGA G	80×25	8×8	640×200	2 (mono) / CGA	Hercules, CGA, PCjr, EGA, MCGA[15][16]
0, 1	0000h, 0001h	VGA Text [17]	40×25	8×8	320×200	16 / CTEXT	CGA, EGA[15]
2	0002h	VGA Text	80×25	8×8	640×200	16 (gray) / CTEXT	CGA, EGA[15]
2, 3	0002h, 0003h	VGA Text	80×25	9×16	720×400	16 / CTEXT	CGA, EGA[15][18]
16	0010h	VGA G	80×25	8×14	640×350	4 / PL4, 16 / PL16	64k EGA,[15] 256k EGA, VGA
17	0011h	VGA G	80×30	8×16	640×480	2 (mono) / PL1	VGA, MCGA, ATI EGA, ATI VIP
23, 88	0017h, 0058h	VGA Text	80×43	8×8	640×350, 640×348	16 / CTEXT	NEL Electronics BIOS, EGA [19]
102	0066h	VESA Text, VGA G, Video7 G	80×50	8×8	640×400	16 / CTEXT, 256K / LINEAR, 256 / LINEAR8	Video7 V-RAM VGA, WD90C, Diamond Speedstar 24X [19]
38, 67, 82, 264	0026h, 0043h, 0052h, 0108h	Video7 Text, VGA G	80×60	8×8	640×480	16 / CTEXT, 256K / LINEAR	Tseng Labs EVA, Tseng ET3000/4000, VEGA VGA, Trident TVGA 8800/8900, Video7 V-RAM VGA, VESA-compatible Super VGA[20][16][19][21][22][23][24][25][26]
35, 20, 23, 27, 39, 65, 2369, 265	0023h, 0014h, 0017h, 001Bh, 0027h, 0041h, 0941h, 0109h	VESA Text, VGA G	132×25	8×14, 9×14, 8×16, 8×8	1056×350, 1188×350, 1056×400, 1056×200	2 (mono) / MTEXT, 4 (gray) / TEXT, 16 / CTEXT, 256K / LINEAR	Tseng ET3000, Tseng ET4000, ATI EGA/VGA Wonder, Cirrus CL-GD5420/5422/5426, VESA-compatible Super VGA [20][16][19][21][22][23][24][25]
29, 66, 84, 86, 266	001Dh, 0042h, 0054h, 0056h, 010Ah	VESA Text, VGA G	132×43	9×11, 8×9, 9×9	1188×473, 1056×387, 1188×387	16 / CTEXT, 256K / LINEAR	VESA-compatible Super VGA
34, 51, 99, 2370	0022h, 0033h, 0063h, 0942h	VESA Text	132×44	8×8, 9×8	1056×352, 1188×352	16 / CTEXT	Tseng Labs EVA, ATI EGA Wonder, ATI VIP, Genoa SuperEGA [15][20][16][19][21][22][23][24][25]
81, 97, 105, 267	0051h, 0061h, 0069h, 010Bh	VESA Text	132×50	8×8	1056×400	16 / CTEXT	MORSE VGA, Cirrus 5320, WD90C, VESA-compatible Super VGA [15][20][16][19][21][22][23][24][25]
33, 82, 30, 268	0021h, 0052h, 001Eh, 010Ch	VESA Text	132×60	8×8, 9×8	1056×480, 1188×480	16 / CTEXT	Tseng ET4000, MORSE VGA, Realtek RTVGA, VESA-compatible Super VGA [15][20][16][19][21][22][23][24][25]

Mode(s) (decimal)	Mode(s) (hex)	Type	Text res. (W×H)	Char. size	Graphics res.	Colors / memory model	Adapters
47	002Fh	Video7 Text, VGA G [27]	160×50	8×8, .	1280×400, 720×512	16 / CTEXT, 256 / LINEAR8	Ahead B (Wizard/3270), VEGA VGA, Genoa [15][20][16][19][21][22][23][24][25]
68, 2372	0044h, 0944h	Video7 Text	100×60	8×8	800×480	16 / CTEXT	Video7 V-RAM VGA, VEGA VGA, Tatung VGA [15][20][16][19][21][22][23][24][25]

VGA and compatible cards support MDA, CGA and EGA modes. All colored modes have the same design of text attributes. MDA modes have some specific features (see above) — a text could be emphasized with bright, underline, reverse and blinking attributes.

The most common text mode used in **DOS environments** and **initial Windows consoles** is the default 80 columns by 25 rows, or *80×25*, with 16 colors and 8×16 pixels large characters. VGA cards always have a built-in font of this size whereas other sizes may require downloading a differently sized font.[28] This mode was available on practically all IBM and compatible personal computers.

Linux kernel 2.6 and later assumes modes *from 0000h to 00FFh* as standard (hexadecimal), if **VGA BIOS** supports, and it understands them as increased by 0x0100. Same for **VESA BIOS** modes *from 0100h to 07FFh* (Linux increases them by 0x0100). Modes *from 0900h to 09FFh* are Video7 special modes, (Usually 0940h=80×43, 0941h=132×25, 0942h=132×44, 0943h=80×60, 0944h=100×60, 0945h=132×28 for the standard **Video7 BIOS**).[29] Linux 2.x allows to check supported video resolutions by kernel argument "vga=ask" or "vga=<MODE_NUMBER>".[30]

Later versions of **Linux** allow specifying resolution by modes *from 1000h to 7FFFh*. The code has a **"0xHHWW"** form where HH is a number of rows and WW is a number of columns. E.g., 1950h (0x1950) corresponds to a 80×25 mode, 2B84h (0x2b84) to *132×43* etc.[29] (Linux 3.x and later allows to set resolution by "video=<conn>:<xres>x<yres>", but it is for video framebuffer graphical mode.[30][31])

Two other VGA text modes, *80×40* and *80×50*, exist but are less common. **Windows NT 4.0** displayed its system messages during the boot process in 80×50 text mode.[32]

Character sizes and graphical resolutions for the extended **VESA-compatible Super VGA** text modes are *manufacturer's dependent*. Some cards (e.g. S3) supported custom very large text modes, like 132×43 and 132×25.[33] Like as in graphic modes, graphic adapters of 2000s commonly are capable to set up an arbitrarily-sized text mode (in reasonable limits) instead of choosing its parameters from some list.

SVGATextMode

On Linux and DOS systems with so named SVGA cards, a program called SVGATextMode[34] can be used to set up better looking text modes than EGA and VGA standard ones. This is particularly useful for large (≥ 17") monitors, where the normal 80×25 VGA text mode's 720×400 pixel resolution is far lower than a typical graphics mode would be. SVGATextMode allows setting of the pixel clock and higher refresh rate, larger font size, cursor size, etc., and allows a better use of the potential of a video card and monitor. In non-Windows systems, the use of SVGATextMode (or alternative options such as the Linux framebuffer) to obtain a sharp text is critical for LCD monitors of 1280×1024 (or higher resolution) because none of so named standard text modes fits to this matrix size. SVGATextMode also allows a fine tuning of video signal timings.

Despite the name of this program, only a few of its supported modes conform to SVGA (i.e. VESA) standards.

General restrictions

VGA text mode has some hardware-imposed limitations. Because these are too restrictive for modern (post 2000) applications, the hardware text mode on VGA compatible video adapters only has a limited use.

<table>
<tr><th>Parameter</th><th>Original VGA</th><th>Modern video adapters</th><th>Remarks</th></tr>
<tr><td>Character cell (glyph) width</td><td>8 or 9 dots[6]</td><td>≤ 9 dots</td><td rowspan="2">Not all hardware support glyphs narrower than 8 dots.</td></tr>
<tr><td>Character cell (glyph) height</td><td colspan="2">≤ 32 dots</td></tr>
<tr><td>Number of character cells</td><td>At least 4,000 (reached at 80×50)</td><td>≤ 16,384 = 2^{14} (memory addressing limitations)</td><td rowspan="3">A modern adapter, if it supports non-standard modes, may produce a reasonably dense text screen even on a large monitor.</td></tr>
<tr><td>Width in character cells (characters per line)</td><td>At least 80</td><td rowspan="2">≤ 256(?)</td></tr>
<tr><td>Height in character cells (number of lines)</td><td>At least 50 (reached at 80×50)</td></tr>
<tr><td rowspan="2">Code page size (number of different glyphs displayed simultaneously)</td><td colspan="2">≤ 512 = 2^9 (if font A ≠ font B)</td><td rowspan="2">Even 512 is insufficient for comprehensive Unicode support.</td></tr>
<tr><td colspan="2">≤ 256 = 2^8 (if font A = font B)</td></tr>
<tr><td>Number of colors</td><td colspan="2">foreground: 16*
background: 8 or 16**</td><td>16 of arbitrarily chosen colors, not fixed.</td></tr>
</table>

* 8 colors may be used by font A and other 8 colors by font B; so, if font A ≠ font B (512 characters mode), then the palette should be halved and a text may effectively use only **8** colors.
** Normally, first 8 colors of the same palette. If blink is disabled, then all 16 colors are available for background.

See also

- General article about *text mode* of computer display

References

1. Petzold, Charles (July 1987). "Triple standard: three new video modes from IBM" (https://books.google.com/books?id=LRBokcwLB70C&pg=PA131). *PC Magazine*. Ziff Davis. Retrieved 13 April 2020.
2. "Appendix D: Console Frame Buffer Drivers" (https://docs.oracle.com/cd/E19253-01/816-4854/euazz/index.html). Oracle. 2010. "On x86 platforms, the Solaris kernel terminal emulator module (tem) uses VGA text mode exclusively to interact with the vgatext module. The vgatext module uses industry standard VGA text mode to interact with x86 compatible frame buffer devices."
3. J. D. Neal (1997). "Hardware Level VGA and SVGA Video Programming Information Page" (https://web.stanford.edu/class/cs140/projects/pintos/specs/freevga/vga/vgatext.htm). Retrieved 13 April 2020. "The corresponding byte in plane 1 is used to specify the attributes of the character possibly including color, font select, blink, underline and reverse."
4. Prosise, Jeff (30 January 1990). "Tutor: modifying character sets" (https://books.google.com/books?id=ySO4VbD0-mcC&pg=PT324). *PC Magazine*. Ziff Davis. Retrieved 13 April 2020. "Unlike IBM's original video adapters, the CGA and the MDA, which store character bitmaps in ROM where they can't be altered, the EGA and the VGA store them in RAM."

5. "The Framebuffer Console" (https://www.kernel.org/doc/Documentation/fb/fbcon.txt). kernel.org. "If fbcon is detached from the console layer, your boot console driver (which is usually VGA text mode) will take over."
6. "VGA/SVGA Video Programming-VGA Text Mode Operation" (http://www.osdever.net/FreeVGA/vga/vgatext.htm). *Osdever.net*. Retrieved 7 November 2016.
7. "Monochrome Display Adapter Notes" (http://www.seasip.info/VintagePC/mda.html). *Seasip.info*. 6 November 2005. Retrieved 7 November 2016.
8. Frank Van Gilluwe (1997). *The Undocumented PC: A Programmer's Guide to I/O, Cpus, and Fixed Memory Areas* (2nd ed.). US: Addison-Wesley Publishing Company, Inc. pp. 172–174. ISBN 978-0-201-47950-8. "Table 6. Sample cursor shapes, Base video port address, Internal mode bits, Screen attribute bit 7 role, Byte for internal mode register at port 3D8h (CGA), 3B8h (MDA) and virtual (EGA/VGA)"
9. "VGA/SVGA Video Programming-Manipulating the Text-mode Cursor" (http://www.osdever.net/FreeVGA/vga/textcur.htm). *Osdever.net*. Retrieved 7 November 2016.
10. Lim, Jeffrey. *Impulse Tracker II User Manual* (https://archive.org/details/ImpulseTrackerIIUserManual). p. 4. "The Tracker runs entirely in text mode with some neat remapping of characters"
11. Lim, Jeffrey (20 March 2014). "20 Years of Impulse Tracker, Part 2" (http://roartindon.blogspot.com/2014/03/20-years-of-impulse-tracker-part-2.html). Retrieved 14 March 2021. "[...]features with some notes:[...] Mouse and character generation functions overall. Text mode let me keep the user interface snappy and memory requirements down, but I bridged the gap with real time character generation beyond what I've seen in other programs."
12. Leonard, Andrew (29 April 1999). "Mod love - Salon.com" (https://www.salon.com/technology/feature/1999/04/29/mod_trackers/index.html). Archived (https://web.archive.org/web/20091124071311/https://www.salon.com/technology/feature/1999/04/29/mod_trackers/index.html) from the original on 24 November 2009. Retrieved 15 March 2021. "Jeffrey Lim, the author of the popular Impulse Tracker program"
13. Lim, Jeffrey (2014). "IT_MOUSE.ASM" (https://github.com/herrnst/impulsetracker). *GitHub*. Retrieved 14 March 2021.
14. Cyrix (16 January 1998). "VGA Function Specification GXm/MXi Processors" (https://web.archive.org/web/20150816220334/http://www.eyetap.org/cyborgs/manuals/soft_vga.pdf) (PDF). Archived from the original (http://www.eyetap.org/cyborgs/manuals/soft_vga.pdf) (PDF) on 16 August 2015.
15. Roschi, Winn L. (1988). "VGA Compatibles: Gaining on the New Standard" (https://books.google.com/books?id=LeosrkjnIM8C&pg=PA177). *PC Mag*. Ziff Davis, Inc.: 177. ISSN 0888-8507 (https://www.worldcat.org/issn/0888-8507).
16. RBIL 61 (INT 10). Set video mode (https://fd.lod.bz/rbil/interrup/video/1000.html)
17. Text
18. Frank Van Gilluwe (1994). *The Undocumented PC* (1st ed.). US: Addison-Wesley Publishing Company, Inc. pp. 319–321. ISBN 0-201-62277-7. "Table 9-2. Video Modes by Adapter Family"
19. Columbia University. Values for standard video mode (INT 10) (http://www.columbia.edu/~em36/wpdos/videomodes.txt)
20. ESTGV RBIL (https://www.estgv.ipv.pt/paginaspessoais/acarv/200607/AC/Docs/interrup.lst.txt)
21. Seabios. Ralf Brown's interrupt list (RBIL) (https://www.seabios.org/Developer_links)
22. The x86 Interrupt List aka "Ralf Brown's Interrupt List" (RBIL) (https://www.cs.cmu.edu/~ralf/files.html)
23. Qemu. Vgabios. vgatables.h (Reference implementation) (https://github.com/qemu/vgabios/blob/v0.4c/vgatables.h#L119)
24. Link to historical Ralph Brown's Interrupt List (RBIL) (https://github.com/balintkissdev/awesome-dos#interrupts)
25. Dosbox official technical info (RBIL, INT10) (https://www.dosbox.com/wiki/Technical_Info#Video_Hardware)
26. Richard F. Ferraro (1994). *Programmer's Guide To The EGA, VGA, and Super VGA Cards : including XGA cards* (https://archive.org/details/programmersguidetotheegavgaandsupervgacardsbyrichardf.ferraro19943rdedition/page/n1231/mode/2up) (3rd ed.). US: Addison-Wesley Publishing Company, Inc. p. 1218. ISBN 978-0-201-62490-8. SBN 201-62490-7.

27. Graphics or text through graphics
28. "OpenBSD Programmer's Manual, vga" (https://man.openbsd.org/OpenBSD-3.9/vga.4). 20 March 1999. "16 different colors can be displayed at the same time. Characters are 8×16 pixels large, and a font consists of 256 characters. A built-in font of this size is always present on a VGA card."
29. Official Linux documentation. 1995-1999 Martin Mares. Video Mode Selection Support (https://www.kernel.org/doc/Documentation/svga.txt)
30. Paul Gortmaker (1999). *The Linux BootPrompt − HowTo, The 'vga=' Argument* (https://www.mit.edu/afs.new/athena/system/rhlinux/redhat-6.2-docs/HOWTOS/other-formats/pdf/BootPrompt-HOWTO.pdf) (PDF). p. 22.
31. *Linux admin guide: kernel-parameters, The 'video=' Argument* (https://www.kernel.org/doc/html/v4.14/admin-guide/kernel-parameters.html). 2017.
32. Olga Kokoreva (2001). *Windows XP Registry: A Complete Guide to Customizing and Optimizing Windows XP* (https://books.google.com/books?id=ApzVAwAAQBAJ&pg=PA223). ISBN 9781931769013. "On obvious difference between Windows 2000/XP and Windows NT 4.0 is the fact that all system messages that appear during the Windows NT 4.0 boot process are displayed in 80×50 text mode, while Windows 2000 and Windows XP display these messages in VGA mode."
33. S3 Graphics. "VC963-3D (S3 ViRGE/DX) User Manual" (https://web.archive.org/web/20200413190535/https://web.aub.edu.lb/pub/os/drivers/VGA/S3VIRGE1/375/VC963-3D.TXT). Archived from the original (https://web.aub.edu.lb/pub/os/drivers/VGA/S3VIRGE1/375/VC963-3D.TXT) on 13 April 2020. "The S3 ViRGE supports 132×43 and 132×25 extended text modes for text applications. This also allows you to emulate terminals requiring 132 columns of text."
34. "Project details for SVGATextMode" (https://web.archive.org/web/20010203143900/http://freshmeat.net/projects/svgatextmode). 19 March 2000. Archived from the original (http://freshmeat.net/projects/svgatextmode) on 3 February 2001. "SVGATextMode uses extra features on SVGA cards to enhance Linux textmodes. It allows setting of the pixel clock, H/V timings, font size, cursor size, etc, and lets you use your video card and monitor to their full potential in textmode."
