Computer Bits

By Hal Chamberlin

UPDATE ON GRAPHICS

WHEN PERSONAL microcomputers were first introduced, input/output facilities were a problem. The norm was a front panel brimming with toggle switches, and rows of binary lights representing memory and register contents. Some experimenters might have purchased a surplus Teletype for external communication with the computer system. But a "television typewriter" with 16 lines of 32 characters was considered to be sophisticated, indeed, while graphics was an unheard of luxury.

In contrast, many of the present crop of packaged and modular computers have CRT display provisions, many capable of displaying as much as 24 lines of 80 characters on the screen. Moreover, graphics capability of some kind is common today, implemented in a variety of ways, and with a wide range of capabilities.

The obvious reason for incorporating video provisions in today's breed of computers is to meet computer users' desires and needs for video. Many computer users, for instance, are interested in increasingly complex, realistic game boards; others wish to pursue creative computer art techniques. Schools form a large block of potential customers, too, with growing interest in graphics for teaching mathematics and physics concepts. Even business applications can use graphics for billing display, financial analysis, word processing, etc.

When selecting a system for a graphics application one must be careful to properly evaluate the capabilities of the systems being considered. There is a large variation in display quality, screen capacity, image resolution, image restrictions, and ease of display programming. For example, photographs of the display screen can be misleading, particularly when in color. As we shall see later, display quality depends heavily on the display monitor and the interface method utilized. Sometimes the images displayed have been painstakingly programmed a bit at a time solely for promotional reasons. Real software support for producing or working efficiently with the image shown may not be available. Also, many graphics systems have severe limitations on the types of images that can be displayed. Therefore, even if the "promotional" video example is possible, other images of similar complexity may not be.

The Display Monitor. If the system being considered does not include its own display (CRT) monitor, the user will have to provide one. The method of connecting a computer to the display is vitally important in determining the resulting image quality.

The least costly and most convenient method is via an r-f modulator connected to the antenna terminals of a television receiver. The r-f modulator simply transmits the computer's video signal on a locally unused channel, via a coaxial cable, to the TV. Unfortunately, in this approach, the display quality is mediocre at best. The sharp cutoff of video frequencies beyond about 3 MHz produced by the sound traps, cause horizontal smearing of characters and images so that 32 characters or 200 dots across is about its limit.

Even if brightness and contrast controls are carefully adjusted for an acceptable picture, the absence of dc-restoration in virtually all TV receivers causes the brightness to shift considerably as screen content changes. The result worsens with systems providing color video.

At this time the r-f modulator needed to effect this interface method is not supplied with the computer. Thus, the user must procure one. A recent FCC directive, however, emphasizes that all such modulators be type approved when combined with a computer since they fall into a Class I device category. Furthermore, the FCC has enjoined r-f modulator makers to desist in selling the devices separately!

The other—and by far preferable method of interface—is direct video to a broadband closed-circuit TV monitor. With this method, r-f modulation/demodulation distortions and the potential for external interference are bypassed. Video frequency response to 10 MHz and beyond is routine, producing sharp edges and consistent brightness to video characters and lines. Broadband monitors are expensive, however, with monochrome units costing from $150 to $300 and color starting at about $500.

Monitors usually have such features as excellent voltage regulation, improved sweep linearity, sharper CRT focus, and dc restoration. When shopping for a used surplus monitor, be sure that the model accepts *composite* video input, since some require separate horizontal, vertical, and video drive signals at TTL logic levels.

A TV receiver can also be converted to a monitor by a knowledgeable hardware person. Although a vast improvement over r-f modulation, the display is not as "clean" as with a true monitor. Usually, an isolation transformer must be added, which increases expense.

Many of the newer computers include a color-display capability. To realize the potential advantages of color, direct video into a color monitor is required. Even so, the display appearance is likely to be visibly worse than that of a monochrome display. Signal degradation is the result of NTSC encoding of the color signal necessary for composite color video. The blues and greens are limited to 1.5 MHz, while red is good only up to 500 kHz. The consequence is that 16 to 20 characters (about 100 dots) is the limit of usable color resolution. Also, unless the monitor is carefully converged, the edges of characters in corners of the screen may be a rainbow of colors, even when color is not being used. The best color systems utilize direct red-green-blue input. Generally, this is only available on a computer system with an integral CRT.

Graphic Generator Techniques. Almost every system or board that does so offers graphics capability in a different way. The wide variety of approaches used is a result of four difficult problems faced by graphics interface designers. The first is *memory usage*, since the image on the screen must be encoded into bytes stored in memory. The second is the amount of graphic detail or *resolution* that it is possible to display. The third is *flexibility*, or the variety of image types

(Continued on page 92)

(Continued from page 87)
that can be displayed. And the fourth is *compatibility* with character generators so that nongraphics applications are simplified.

These are conflicting requirements. For example, a high-resolution flexible display will require considerable memory to store the image. On the other hand, maintaining compatibility with character generators minimizes memory usage but limits resolution and flexibility.

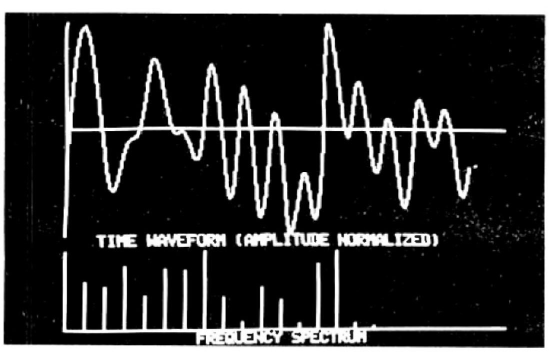
In most cases, designers have opted for compatibility with character-oriented displays. Such a display divides the screen into rows and columns of *character cells*. The PET computer for example, uses 25 rows of 40 characters per row for a total of 1000 character cells. Each cell, in turn, is divided into *pixels* or dots; the PET uses 8 rows of 8 dots per row. The characters that make up the display are therefore displayed centered in the character cells and are, in turn, composed of the dots. Each byte of display in the PET memory corresponds to a character cell. Hence, 1000 bytes (1 KB) of memory are used for the display.

Graphic images can be formed from a special *graphics character set* used in addition to the normal ASCII character set. Shapes such as horizontal, vertical, diagonal, and curved lines are provided and can be pieced together to form crude line drawings. Special symbols, such as hearts, clubs, spades, etc., are also included for a total of 64 graphics characters. Add the 64-character uppercase ASCII subset and a bit for inverse video (black-on-white) and all 256 possible character codes are used up.

Such a display is very adept at putting on-screen gameboards, simple line drawings, and, of course, text. On the other hand, mathematical curve plotting resolution is no better than the typical character plot with "*" for points. Although one may be able to hand-fit a curve with the line segments and other shapes available, there is no software support available to do this. Moving images can also be programmed, but the movement is quite choppy since it must be in character cell increments.

A related technique employs a *programmable* character generator, such as one made by Objective Design for S-100 bus compatible display interface boards. Essentially, the usual ROM character generator is replaced by a small RAM that can be written into to provide a changeable character set. In the Sorcerer from Exidy, for example, the character to be displayed can be determined by software. Thus, it is now possible for a user to tailor the graphic character set to match the application. If the application is line drawing, the RAM can be filled with a greater variety of line segments and curves. If it is chess, chesspiece symbols can be written. Whatever graphic set is chosen, it must be used for the entire display. Even with the added flexibility, the limitations of this scheme are basically the same as with the fixed-graphic character set.

Another variation attempts to increase the screen resolution for random-dot graphics such as required for curve plotting. Radio Shack's Model TRS-80 display, for example, uses 16 rows of 64 character cells per row with a character cell being 12 rows of 8 dots each. For graphics, each character cell is divided into a two-wide by three-high array of blocks or pixels and any possible on/off combination of the six blocks can be specified. In the graphics mode, therefore, the screen becomes an array of 48 rows of 128 *blocks* per row and any conceivable combination of blocks can be on and off. While the resolution for line drawing and game boards is inferior to the graphic character generator approach, arbitrary curves are displayed with about twice as much resolution. Display memory in the TRS-80 is only 7 bits wide and comprises 1024 bytes. Normal ASCII characters take up 64 of the codes, while the other 64 codes are used to specify the 64 possible combi-



The Micro Technology Unlimited Visible Memory in action.

nations of graphic blocks in a character cell.

Bit-Mapped Displays. An entirely different approach to graphics ignores character generators altogether and simply divides the screen up into a very large number of individual dots, with one bit of display memory for each dot. Such displays are called *pixel* or *bit-mapped* displays to distinguish them from character displays. If a sufficient number of dots are provided, this is by far the most flexible graphic display technique because there are no restrictions on image type or placement.

When using the proper software, even text can be displayed with complete freedom as to character shape, size, and placement. This makes possible bold headlines, tilted italics, subscript and superscript for math and chemical equations, and proportionally spaced text for clean right-margin justification. Their main disadvantage is that a large amount of memory is needed for a high resolution display. Upwards of 8K bytes is not uncommon.

The Apple II computer, for example, has a black-and-white pixel display mode with 192 rows of 240 dots per row, which is sufficient resolution for well detailed drawings, graphs, and charts. The 120 by 96 full-color mode, which can display 7 colors at two intensity levels and black, is less useful for curve plotting but is capable of beautiful computer art.

Users of KIM-1s can use the Visible Memory (see photo), which provides a 320-wide by 200-high dot matrix as well as 8K of refresh memory, all on one board. Owners of S-100 bus micros can utilize a new graphics interface from Vector Graphics that provides a 256 by 240 image in black and white or a 128 by 120 image with 16 levels of gray. The latter mode is useful for half-tone picture processing and display in applications such as amateur slow-scan TV.

Software support is important in any graphics application, particularly if only BASIC is used. Although the user can perform any graphic operation by POKEing data into memory, BASIC is likely to be very slow in manipulating the large amount of data required. Radio Shack, on the other hand, provides BASIC statements for setting and resetting any graphic cell given its *X* and *Y* coordinates. The Apple provides statements for setting the coordinates of endpoints and will automatically and rapidly draw the best straight line connecting them. ◇