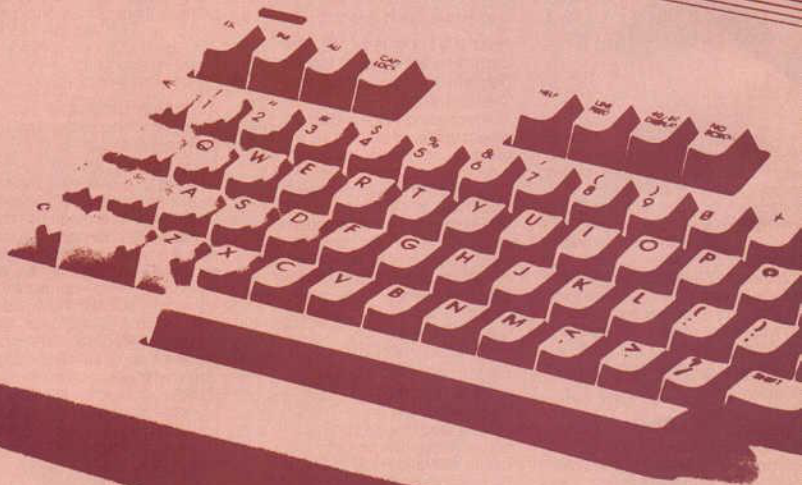


INSIDE THE 128

The new Commodore 128 Personal Computer has generated quite a bit of interest, especially by current owners of the popular Commodore 64. Is the 128 a significant enhancement, or just a warmed-over 128K version of the 64? We went to Commodore's headquarters to find out, and came back with a new appreciation for this intriguing machine.

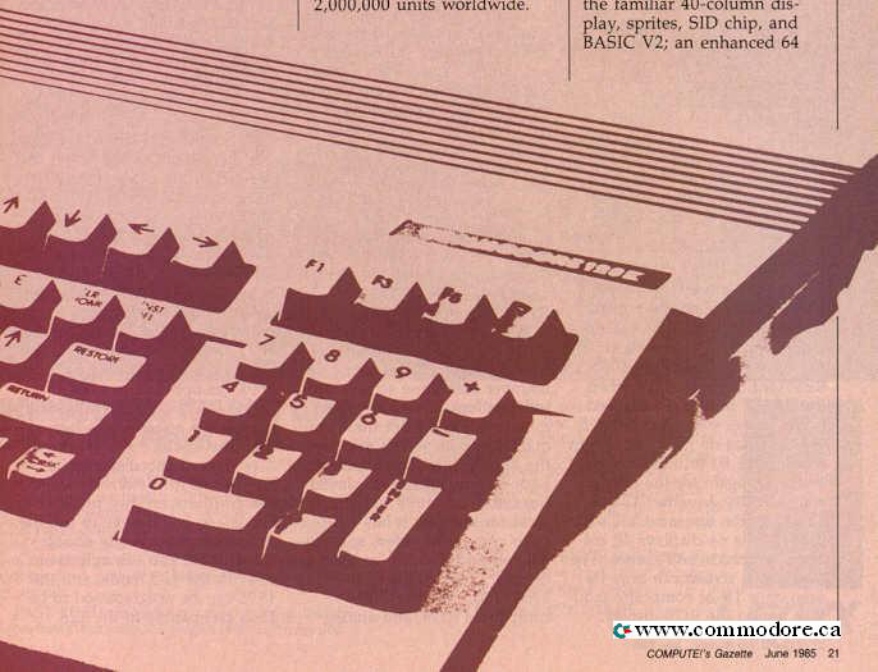


A Hands-On Look At Commodore's Newest Computer

Charles Brannon, Program Editor

Soon after it was introduced, the Commodore 64 proved to be the leader of a new wave of home computers. Even at the original price of \$600, the 64 came equipped with as much memory as \$2000 business machines, along with arcade-quality graphics, detailed animated sprites, and a unique sound synthesizer that brought realism to what was formerly just beeps and tones. The 64 became one of the most popular computers ever, selling over 2,000,000 units worldwide.

The 64 is firmly established, with over 6,000 programs to its credit. But as good as the 64 is, we've been waiting three years for an encore. Although it's been high time for an enhancement, no one wants to give up his or her personal software library. Commodore's answer, the Commodore 128 Personal Computer, provides true 64 compatibility, plus a real advance in power and flexibility. The Commodore 128 is literally three computers in one: a Commodore 64 with the familiar 40-column display, sprites, SID chip, and BASIC V2; an enhanced 64



RGB (for use with the Amiga). With the 1902, you must manually flip a switch after you change screen modes.

This can be cumbersome, but Commodore feels that you'll probably stay in one mode or the other, a reasonable assumption. This scheme does let you have two simultaneous displays. Perhaps one screen could show color graphics, while your program listing is displayed on another. One can envision dual-perspective games with each player having his own independent screen.

The 1902 composite/RGB display will probably sell for under \$400. The least expensive route, though, is to use a television for 40 columns, and a monochrome (black and white) monitor for 80 columns. Commodore will sell a special cable to connect the RGB port to a monochrome monitor. The cable can be used with Commodore's inexpensive 1901 monochrome display and with other monochrome monitors.

The new 1571 disk drive further amplifies the power of the 128. In 64 mode, the 1571 behaves just like a 1541. The 1571 we worked with was not quite ROM-compatible with the 1541 (our "TurboDisk" program did not work with it), but we were assured that 1541 compatibility, a high priority, was being improved. In the 128 mode, the 1571 shows its true power, boosting storage capacity to 360K (as opposed to 170K on the 1541), and transferring data from seven to ten times faster than the 1541.

The enhanced storage is due to the 1571's double-sided design (there are two read/write heads), so you'll have to use the somewhat more costly double-sided disks. You can still use a 1541 in the 128 mode, and the 1571 can be programmed to be 1541 compatible in the 128

with 128K and all 64 features, plus 80 columns and BASIC 7.0; and a true CP/M-compatible machine, promising the ability to run off-the-shelf CP/M software. And all at a price almost anyone would call reasonable: under \$300.

Compared to the 64, the 128's console is much bigger, perhaps to imply more power, but probably necessary to hold the hardware of three computers. The main part of the keyboard is identical to the 64's, except that the function keys have been moved to the upper-right corner and rearranged horizontally. There is a numeric keypad with +, -, ., and an ENTER key (synonymous with the RETURN key). Along the top of the keyboard are ESC, TAB, ALT, CAPS LOCK, HELP, LINE FEED, 40/80 DISPLAY, and four separate cursor keys.

None of these additional keys, not even the keypad or separate cursor keys, function in the 64 mode, for the sake of true compatibility. Adding extra programming in ROM to support these keys in 64 mode might be just enough to prevent some 64 software from working properly. Commodore is staunch on this; anything less than 100% compatibility isn't good enough.

In the 128 mode, the 40/80 DISPLAY key selects which screen mode is used as the default. This key is checked at power-on, when RUN/STOP-RESTORE is pressed, or when the RESET button (found next to the power switch) is pressed. This key has no meaning in 64 mode since 80 columns are not available, again for the sake of compatibility. In either 128 or CP/M mode, the same VIC chip used on the 64 displays 40 columns, graphics, and sprites. The 40-column screen can only be seen on a TV or composite monitor, not on the RGB display.

The RGB monitor displays twice as many pixels and characters as 40 columns, and achieves color purity since the signal is separated into the red/green/blue color components. (A composite signal has all the color information mixed together, which makes it difficult to cleanly separate these colors.) A special video chip is used for 80 columns. The 80-column screen can only be seen on the RGB monitor. All 16 colors are available in 80 columns (although the Commodore-1 color, normally orange, appears as dark purple), as well as reverse video and underlining. Unlike the 40-column mode, there are 512 characters available in 80 columns, which means you can get both uppercase, lowercase, and all keyboard graphics simultaneously.

This 80-column chip is for text only—it does not support bitmapped graphics or sprites. You can redefine the character set, though, and set up a small 640 × 48 simulated bitmapped window. The 80-column video chip uses 16K of dedicated screen memory. None of the 128K memory is used for 80 columns, so in effect this machine actually has 144K of total RAM.

There are three ways to switch between 40 and 80 columns: toggle the 40/80 switch and press RUN/STOP-RESTORE, press ESC-X in BASIC, or enter the command SCREEN 0 for 40 columns, or SCREEN 5 for 80 columns. Remember that these screens are independent. If you have two monitors hooked up, these commands reroute screen printing to the appropriate screen (although both screens remain displayed). Commodore's 1902 monitor is ideal for the 128; it has built-in color composite video, split signal composite video (as used on the rear connections of the 1701/1702 monitor), IBM-compatible RGB, and analog

can examine the error number in the reserved variable ER, the number of the line that caused the error in EL, and the error message with the function ERRS. After you've handled the error, RESUME returns control to the statement after the error, or to any line number.

Text processing is enhanced with INSTR, which finds the position of a substring within a larger string. PRINT USING lets you define a format field for printing, making it easy to set up columnar tables and forms. WINDOW sets up a smaller screen that scrolls independently from the rest of the screen. WINDOW can be used to emulate simple Macintosh-style windowing.

No more POKES for SID chip sound. BASIC 7.0 includes several commands for music and sound effects. SOUND sets the frequency, duration, and waveform of a sound effect. You can also specify a sweeping effect. PLAY is a mini-language of its own. You can use it to play strings of notes, specifying note names, durations, sharps/flats, dotted notes, and rests. You can use it to synchronize three-voice music, set the filter, and control individual volume for each voice. Each voice can play from a set of predefined envelopes that simulate one of ten musical instruments: piano, accordion, calliope, drum, flute, guitar, harpsichord, organ, trumpet, and xylophone. You can customize these preset instruments with ENVELOPE, customize the programmable filter with FILTER, set the overall VOLUME, and the TEMPO of music.

BASIC 7.0 offers a rich vocabulary of graphics commands. GRAPHIC is used to enter either the multicolor 160 × 200 graphics screen, the hi-res 320 × 200 graphics screen, the 40-column text screen, or the 80-column text screen.

GRAPHIC allow you to define a text window and can either clear the screen or leave previous graphics in place. SCNCLR can also be used to clear the screen. When you enter a graphics mode, the start of BASIC is moved beyond the end of the graphics screen. GRAPHIC CLR is used to deallocate the memory used by the graphics screen. RGR returns the number of the current graphics mode.

DRAW is used to plot a single point, or draw a single or a connected line to create complex shapes. LOCATE is used to set the position of the graphics cursor without plotting any point. BOX can draw any rectangle or filled rectangle, at any angle. CIRCLE is used to draw circles, ovals, arcs, or any polygon, at any angle of rotation. You can place text anywhere on the graphics screen with CHAR. You can also use CHAR on the text screen to simulate PRINT AT. COLOR is used to set any of the color registers, and the function RCOLOR reads which color is assigned to a color register. PAINT can fill any shape with any color. SSHAPE can "pick up" any block of the screen and store it in a string. This shape can then be copied back to any place on the screen with G SHAPE.

A pixel can be tested with the function RDOT, which returns the color of the pixel at the specified row and column. The WIDTH command specifies the size of pixels plotted. A WIDTH of 2 makes all lines double-wide. And finally, the SCALE command lets you pretend that the screen is actually 1024 × 1024 pixels across and down. You can use this range in your drawing statements, and the coordinates are automatically scaled to fit the actual screen size.

BASIC 7.0 just wouldn't be complete without sprite commands. If you've been stymied

by POKE and PEEK for sprite control, as well as the infamous "seam," you'll really appreciate the following sprite commands.

First, BASIC 7.0 includes a simple sprite editor. Just type SPRDEF, and a box appears on the screen. Enter which sprite you'd like to define, then use the cursor keys and the number keys 1-4 to draw squares on the grid. When you're through, the sprite is stored into a reserved section of memory. This memory can be saved to disk with BSAVE, then recalled within your program with BLOAD, eliminating the need for DATA statements.

To set up sprite parameters, use SPRITE. This command turns on the sprite, sets its color, priority, initial X and Y position, and sets hi-res or multicolor for that sprite. You can then use MOVSPR to position the sprite anywhere on the screen. MOVSPR can also be used to set the sprite into motion. After you specify the speed and angle, the sprite moves on its own. Your program continues in the meantime. (Sprites are updated in this mode during the IRQ interrupt.) While a sprite is in motion you can read its position with RSPPOS. You can transfer the sprite pattern into any string, or copy a sprite pattern from a string into any sprite. In combination with SSHAPE and GSHAPE, you can "pick up" a block of the screen and turn it into a sprite, and "stamp" the sprite pattern anywhere on the graphics screen.

SPRCOLOR sets the multicolor registers shared by all sprites and the function RSPRCOLOR reads the sprite multicolor registers. The COLLISION statement transfers control to a specified line number when two sprites touch, or when a sprite touches part of the screen background. Your collision routine can see what

caused the collision with the function BUMP.

No longer are PEEKs, POKEs, or machine language necessary to read the game controllers. The function JOY returns the status of either

joystick. POT returns the position of one of the four paddles, and PEN is used to read the X,Y coordinates of the light pen.

A few miscellaneous commands: SLEEP is used as a delay loop, pausing from 1 to

65535 seconds. GETKEY is like GET, but waits for a keystroke. GOM4 exits to the 64 mode, but first asks ARE YOU SURE?, since anything in memory in the 128 mode will be lost.

The BASIC 3.6 in the Com-

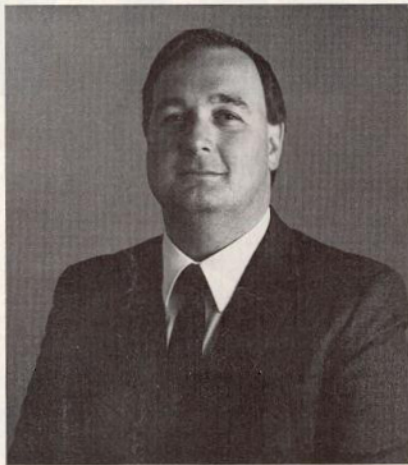
modore LCD lap portable can run BASIC 7.0 programs as long as the programs shy away from PEEKs and POKEs (which shouldn't be needed very much with such a powerful BASIC). Although the LCD lacks sprites and the SID chip, its BASIC is programmed to merely ignore the BASIC 7.0 commands it can't execute, instead of crashing with a SYNTAX ERROR. This is indicative of a possible Commodore plan to pair these machines. The LCD portable would be used to acquire data in the field, which could then be uploaded to the 128 for further processing.

BASIC 7.0 has almost every command a programmer would need. There are almost too many commands, extending the time it takes to learn a programming language. However, you need not memorize every command, just learn commands as you need them. You'll want to at least be aware of the available commands, though, so you won't reinvent the wheel by POKEing your way to sound or graphics.

The 128 is a logical upgrade of the 64. Without sacrificing 64 compatibility, the 128 answers almost anyone's wish list. BASIC 7.0 gives programmers freedom to program without POKEs or cumbersome machine language routines. The 80-column display, 2 MHz microprocessor, 128K of memory (theoretically expandable to a megabyte), CP/M Plus, and fast double-sided disk drive make the 128 a capable business machine, competitive with the much more expensive IBM and Apple computers. As usual, though, we'll still have to wait for software to be written that takes advantage of these features. Although you can use existing 64 and CP/M programs, it looks like you'll have to write your own 128 mode programs for a while.

An Interview With Paul Goheen Commodore's Director of Software

Selby Bateman, Features Editor



Paul Goheen, who directs Commodore's software division, can be considered one of the genuine veterans in the post-Jack Tramiel Commodore environment. He joined Commodore during the latter part of 1980, shortly after the launch of the 8032 computer and before the introduction of the VIC-20. Previously a programmer, systems analyst, and consultant in the mainframe and minicomputer arenas, Goheen came to Commodore as a software product manager at a time when Commodore's U.S. computer operations included only about 20 people. In the past few years, he has seen the amazing growth and the many changes which have occurred at the company and in the microcomputer industry in general.

When COMPUTE!'s GAZETTE recently spoke with Goheen, Commodore was working full tilt to launch the 128 into the retail market by May and June. Also underway was the surprising lap portable computer now planned for an early fall release. Both the 128 and the portable were introduced at the January Consumer Electronics Show (CES). In the wings is the Amiga Lorraine computer, an eagerly awaited powerful and versatile 16/32-bit machine which Commodore has kept under wraps since purchasing the Amiga company last summer. The Amiga is scheduled for a midsummer entrance.