

B+ Advanced BASIC

By: Jon Mattson

Published by: Loadstar, Issue #66, 1989

Presented by: www.My64.in.nf

B+ adds 36 new commands to the BASIC language. Some let you avoid annoying PEEKs and POKEs; others let you use special effects normally available only to machine language programmers. And, best of all, you don't require any knowledge of ML to use it!

Using B+ is easy. First, load the utility into memory in direct mode, type: **LOAD "B+",8,1:NEW.**

To use it in a program, add the following line to the very beginning of your program:

```
10 IF A=0 THEN A=1: LOAD "B+",8,1
```

Next, enter **SYS 49152**, either in the program or in direct mode, and the title screen will appear (similar to the top of this page). Now you can use all of the new B+ commands, just like normal BASIC! Programs which contain B+ commands can be LISTed, LOAded, SAVEd and RUN normally, as long as B+ is in memory.

Some things to keep in mind while using B+:

Hit the RESTORE/STOP combination as often as you like! B+ will stay put unless you tell it to QUIT. Note that its color registers will also stay in whatever format you have specified, so say good-bye to that often hard to read blue-on-blue combination (unless, of course, you specify it yourself).

B+ fills the empty memory block in 49152 to 53247, and uses 820 to 1023 for various pointers, so don't POKE to these areas. The 'hidden' memory beneath 40960-49151 and 57344-65535 is used to store screens, windows, etc., as well. Note that this means that, for all of its many functions, B+ doesn't take 1 byte away from your BASIC storage area.

The B+ commands are fully integrated within the BASIC interpreter with ONE EXCEPTION. Since the interpreter does not check for new commands within an IF...THEN clause, you must 'fool' it into thinking the command begins a new statement. This can easily be done by adding a colon, e.g.

```
IF A=1 THEN: COLOR 0,A, instead of
```

```
IF A=1 THEN COLOR 0,A
```

This has no effect on the syntax or logic of the statement (even in normal BASIC).

As with normal BASIC, B+ commands often call for one or more parameters - e.g. in "COLOR1,2", 1 and 2 are parameters. A standard system has been used for frequently called parameters - entering a number below or above the accepted range will generate an "ILLEGAL QUANTITY ERROR", as usual. These common ranges are:

Sprites	0 to 7
Colors	0 to 15
Screen Columns	0 to 39
Screen Lines	0 to 24

As usual, parameters can be constants (1,56), variables (A,DE,X\$) or any reasonable combination thereof (A+1, 2*D+G/2).

*** THE COMMANDS ***

Each command is listed in a standard format below. Required parameters are noted in small letters after each command word title. Unless otherwise noted, all parameters are mandatory. References to related commands may also be noted.

BANK number

Changes the current 16K video bank to a number between 0 (normal) and 3, representing banks starting at 0, 16384, 32768 or 49152 respectively.

See also CSET & SCREEN.

BLANK

Blanks the screen to the border color or 'un-BLANKs' it from a BLANKed condition (i.e. it acts as a toggle).

BLOAD "filename",sa

Loads a file into a specified memory address, 'sa'. May be issued within a program, avoiding the old "IF A=0 THEN A=1: LOAD..." routine.

BSAVE "filename",sa,ea

Saves a section of memory to disk, starting at address 'sa' and ending at address 'ea'. Note that 'ea' should be 1 higher than the actual end of the section, i.e. BSAVE "MYFILE",32768,34816 saves the memory from 32768 to 34815.

CHAR column,line,"text"

Moves the cursor to the specified column and line, then places "text" in that spot.

CLS

Clears the current screen and returns the cursor to the 'home' position.

COLOR register,color

Changes the color in the appropriate register:

0	Border	(53280)
1	Background 0	(53281)
2	Bkg 1 (Multi 1)	(53282)
3	Bkg 2 (Multi 2)	(53283)
4	Background 3	(53284)
5	Sprite Multi 1	(53285)
6	Sprite Mutli 2	(53286)
7	Text	(646)

Note that any color specified with this command becomes the default color for that register (i.e. hitting the RESTORE/STOP combination will not change it).

COPY set,location

Copies the character set (0 for normal, 1 for lower case) to the specified memory location. Note that this is normally only useful if you copy it to an even 2K boundary. This command only copies the set - it does not display it.

See also BANK, CSET & SCREEN

CRSR column,line

Moves the crsr to the indicated column and line.

CSET location

Changes the character pointer so that you can use different character sets. The location should be an even 2K boundary (normal value = 4096). Make sure that you are in the correct 16K video bank to view the new set.

See also BANK,COPY & SCREEN

NOTE: CSET can also be used to alter the location of a high resolution screen (see MODE).

CWINDOW

Closes an open window, returning the screen to its previous state.

See also OWINDOW, WCLR & WINDOW

DICE range, variable%

Generates a random number within a specified range (2 to 128) and places

it in the indicated INTEGER variable. For example, DICE 6,D% will act like a normal 6-sided dice, placing a number from 1 to 6 in D%.

DIRECTORY

Lists the current disk directory to the screen, WITHOUT wiping out your BASIC program. No more LOAD"\$",8!

FCOL color

Fills the current screen with the specified color.

FILL character

Fills the current screen with the specified character.

HOME

Moves the cursor to the 'home' position without otherwise affecting the screen.

JOY stick,variable%

Checks the specified joystick (1 or 2) and returns the reading in the chosen INTEGER variable (e.g. J%). A value of 0 indicates that the stick is not being moved. A value of 1 to 8 means that it is being pressed in the direction indicated below:

```
      1
    8   2
  7   0   3
    6   4
      5
```

A value of 128 or higher indicates that the fire button is being pressed: subtract 128 from the result (or AND it with 127) to find the direction in this case.

LCLEAR first,last

Clears individual screen lines. For example, LCLEAR4,7 will erase all of the lines between 4 and 7, inclusive. The cursor is then moved to the top of the created space.

LINK stick [,sprite,speed]

Links the specified joystick (1 or 2) to the indicated sprite (0 to 7), so that moving the joystick will move the sprite appropriately. The speed can be 1-255 (0 counts as 255), with 1 being fast and 255 being very slow (3 to 5 is normal). Typing LINK 0 with no sprite and speed parameters will break the connection.

Note that this routine is interrupt driven: it will function even while another program is running! Thus, it is very useful for creating GEOS-type pointers and such. As with other interrupt-driven routines, it is a good idea to turn off LINK when accessing the disk drive or using the RASTER command (below).

MODE number

Changes the type of screen in use:

- 0 Normal
- 1 Multi-color
- 2 Extended Background
- 3 High Res
- 4 Multi-color High Res

The location of a high resolution screen must be specified in the usual manner or by using B+'s BANK, SCREEN and CSET commands. BANK specifies the video bank, and SCREEN specifies the video matrix within that BANK. CSET allows you to toggle between high res screen 0 and 8192, since it affects bit 3 of 53272. The easiest way to use it is simply to specify the CSET parameter as being the same as the beginning of the high res screen.

Ex. "BANK1:CSET24576:SCREEN23552:MODE3" will set up a normal high res screen from 24576 to 32767, with the video matrix at 23552.

See also BANK, CSET, PLOT & SCREEN

MOVSPR sprite,x,y

Moves the indicated sprite to the specified x and y coordinates. The usual ranges are allowed: 0-343(+) for x and 0-255 for y (although not all of these positions are visible).

OWINDOW "text"

Opens a window on the screen which contains the specified text string (color characters may be embedded in it to alter the color of the text). This also saves the screen below the window so that it will be restored when the window is closed. The window attributes (location, size, etc.) are determined by the WINDOW command. Note that windows can be "re-used" by issuing this command more than once in succession with different text strings, without specifying WINDOW parameters in between. Since there is a limit of 32 window parameters at a time, this avoids using them up needlessly. Closing the last of these over-lapping windows will restore the screen beneath the pile (not the previous windows).

See also CWINDOW, WCLR & WINDOW

PAINT character,color

Instantly scans the screen for every occurrence of the specified character and changes each one's color as indicated. This command can be issued for several different characters to rapidly "colorize" a screen.

PLOT mode,x,y

Used to plot points on the current high resolution screen at the noted x and y coordinates. Mode specifies the type of plot desired:

- 0 Clear pixel (erase)
- 1 Set pixel (draw)
- 2 Reset pixel

Resetting a pixel clears it if it is set or sets it if it is clear (i.e. an "exclusive or" condition).

See also MODE

QUIT

Exit B+ - you may restart it with a SYS 49152 as long as it is in memory.

RASTER area,c0,c1,c2,c3,c4,mode,multi,bank,screen,cset

B+ allows you to create raster-driven split screens which can be controlled from BASIC. This is one of its most powerful commands, but also one of its most sensitive. RASTER lets you set the parameters for one area of the screen (0 for top, 1 for bottom); thus, it is usually issued twice, once for each half. Parameters c0 to c4 represent the colors, as per the COLOR command (i.e. border and background 0 to 3). Other parameters include:

- mode 0-2 normal/extended bkgrnd/high resolution
- multi 0-1 normal/multicolor
- bank 0-3 as BANK command
- scrn 0-15 1K screen area within 16K bank: actual
area = 64 x bank + 16 x scrn
- cset 0-14 2K character set area within 16K bank: should
be an even number

For those who are interested, the parameters represent the following memory locations, respectively: 53265,53270, 56576, 648, 53272. The normal values are 0,0,0,1,4. Note that RASTER does not initiate the split screen - it only sets up the parameters for it.

See also RASTON & RASTOFF

RASTOFF

Turns off the current raster split screen. This command should be issued during disk access, including use of the DIRECTORY command.

See also RASTER & RASTON

RASTON line

Activates the previously specified raster interrupt, splitting the screen at the indicated line (0-25 in this case). The split screen should always be shut off (RASTOFF) during disk access.

See also RASTER & RASTOFF

RECALL screen

Recalls a previously STOREd screen, specified by a number between 0-3.

See also STORE

SCREEN location

Moves screen memory to the specified location, which must be an even 1K amount. May also be used to move the video matrix of a high resolution screen. Remember that the screen will only show up if it is in the correct video BANK.

See also BANK, CSET & MODE

SPRITE number, on/off [, color, priority, x-expand, y-expand, multicolor]

Turns off the sprite (e.g. SPRITE3,0) or turns it on with the indicated parameters (eg. SPRITE3,1,4,0,1,1,0). Color alters the basic color of the sprite. Priority indicates whether it will appear in front of (0) or behind (1) screen characters. X-expand, y-expand and multicolor are toggled on with 1 and off with 0.

STORE screen

Saves the current screen and colors to the specified buffer, 0 to 3. The screen may then be restored, entirely intact, later on with a RECALL.

See also RECALL

WCLR

Clears the current window pointers. This will free up room for more windows, but will not allow you to close any which are already open. Since up to 32 windows can be open at once, this should not normally be necessary unless you wish to leave windows permanently open.

See also CWINDOW, OWINDOW & WINDOW

WINDOW column,line,width,height, border character,border color

Specifies the parameters for a window to be opened with OWINDOW. Column and line specify the upper left corner of the window. Width and height specify its size in characters. Of course, column + width cannot be greater than 40, and line + height cannot exceed 25. Border character specifies the screen code of the character that will be used to draw the border; a 0 here will draw a proper line box using a combination of characters. Border color specifies the color of the window "frame", not the text within it. Up to 32 windows may be specified and opened at once, but their total area cannot exceed four screens (4000 bytes). Exceeding either of these limits will generate an "OUT OF MEMORY" error. Remember that "re-using" a window without resetting its parameters does not eat into this limit (see OWINDOW).

See also CWINDOW, OWINDOW & WCLR

WIPE

Clears the high resolution screen - make sure that you have one in effect before issuing this command! This does not clear the video matrix: use either CLS or FILL0 to do this.

See also MODE

***** Hints & Tips *****

RASTER Split Screens

This group of commands allows you to do some pretty weird and wonderful stuff, even in BASIC; however, a price comes attached to that power. Keep in mind the following factors when using a split screen:

- The computer's processing speed can't always keep up with the demands placed on it. As C-128 owners will already have discovered, the bottom line just below a split screen has a life of its own. A sprite crossing this boundary can cause all sorts of (temporary) pyrotechnics - even pressing keys can cause it to jiggle. If the two halves have different character sets or modes (i.e. low and high res), this becomes even more annoying. The solution: avoid using the offending line. Don't put text on it if you can help it and try to avoid crossing it with sprites.

- Anything which detracts from the servicing of interrupts can screw up the split screen. This includes: disk access, the LINK command and frequent key presses. Never mix RASTER with any of these if you can help it.

- Remember that you can freely alter the raster split to achieve all sorts of fascinating effects. For example, a "winking" screen can be created with this simple line:

```
FORI=0TO25:RASTONI:NEXT:FORI=25TO0 STEP-1:RASTONI:NEXT:RASTOFF
```

- Avoid using screen/bank altering commands when a complicated split screen is in effect: since you have no way of knowing which half of the screen the raster is working on at that precise moment, the effects can be rather unpredictable. Similarly, remember to "clean up" the screen after a split is turned off by issuing the desired screen commands: you have no way of knowing which half of the screen became the whole thing when the raster was deactivated.

- For Experts Only: B+ uses shadow registers to hold the 10 parameters

required by the RASTER command. By altering these registers directly, you can get interesting results not possible with the simplified RASTER choices (i.e. screens which scroll smoothly at two different speeds, half-blanked screens, etc.). These registers are located at 50278-50287 for the top half of the screen and 50288-50297 for the bottom. The order is the same as for the RASTER command.

WINDOWing

For those who require more than 32 windows(!), try mixing the WINDOW and STORE/RECALL commands and see if that helps. In effect, you have four more (full screen) windows at your disposal with the latter commands. If you require only four or fewer screens to be saved you can actually have up to 128 open windows, by alternating STORESs with the opening of 32 windows. And remember that the OWINDOW command can be used forever if you re-use the same parameters, since the screen underneath need only be saved once.

- * Remember: The best way to learn something is to try it yourself! Experiment with the commands, examine the Demo, and soon you won't even need this manual! *