

S T A R B A S I C

By: Steven Emsley
Published by: Loadstar, Issue #90, 1991

Presented by: www.My64.in.nf

To install STAR BASIC, simply LOAD"STAR BASIC",8,1 and activate it with a SYS 35792, followed by CLR. If disabled, either with the QUIT command, or a RUN STOP/RESTORE, it can be re-enabled with a SYS 51200.

STAR BASIC identifies the new keywords by using an exclamation point (!) in front of them. This can be changed to another character if preferred by POKEing the PETASCII value of the character desired into location 51304. For instance if you'd prefer the pound sign then you could use the following POKE:

```
POKE 51304,ASC("#")
```

STAR BASIC uses the following memory locations when in operation. \$8C00-\$9FFF (35840-40959) and \$C800-\$CFFF (51200-53247). In addition, a few commands also use \$E000-\$E7FF (57344-59391) underneath the kernal.

The normal amount of BASIC bytes free is 38911. With STAR BASIC in place, it drops down to 33791. Also, STAR BASIC may not work in conjunction with other programs, as it redirects up to 4 vectors for its own purpose.

In the following examples, the numbers in parentheses are necessary parameters. Anything in brackets are optional. In addition, the values for ROW are 0-24, for COLUMN they're 0-39, and for COLOR they're 0-15.

----- **BASIC ENHANCEMENTS** -----

RESTORE - allows the resetting of data pointers to a specific line number besides the beginning.

Usage: !RESTORE [line number]

GOTO - allows variables instead of line numbers.

Usage: !GOTO (line number, variable, formula)

GOSUB - allows variables and formulas.

Usage: !GOSUB (line number, variable, formula)

NOTE: These enhancements are for GOTO and GOSUB only, and not ON-GOTO or ON-GOSUB. Since variables and formulas are now allowed, the ON command is no longer needed.

EXAMPLE:

```
ON X GOTO 100,200,300,400,500,600
```

can be redone as the following:

```
!GOTO X*100
```

GET - emulates the C-128 GETKEY command, so...

```
10 GET A$:IF A$="" THEN 10
```

can now be done as

```
10 !GETA$
```

If used with a variable instead of a string, the numbers 1 thru 9 will set the variable to the appropriate number, while all other keys will set the variable to a zero. You don't have to worry about a crash if someone presses a non-numeric key.

Usage: !GET (variable)

INPUT - filters out unwanted keypresses, such as CRSR keys, function keys, HOME key, etc. Also avoids the EXTRA IGNORED error by allowing the comma and semicolon. A length of the input can also be set, but it has to be a number value, and not a variable value. Additionally, it is set up for string variables only.

Usage: !INPUT (string variable),[len]

NOT ALLOWED:

!INPUT R

Must be string variable. Equivalent would be !INPUT R\$:R = VAL(R\$)

!INPUT R\$,L - where L=Limiter must be >0 and <255.

----- GENERAL COMMANDS -----

QUIT - this exits STAR BASIC, and resets all the pointers that it used.

Usage: !QUIT

REPEAT - toggles repeating of all keys or the normal ones, such as SPACE, CRSR keys, etc.

Usage: !REPEAT

FILL - fills a section of memory with a specific byte.

Usage: !FILL (start), (end+1), (byte)

!FILL 4096,8192,0 - fills from 4096 to 8191 with 0.

CFILL - the same as the FILL command, but is specifically for color memory of 55296-56295.

Usage: !CFILL (color)

MOVE - moves a section of memory. This command is set up for the move to be in 1 of 3 formats:

1. moving memory forward
2. copying itself
3. moving memory higher, but the new start must be higher than the end.

Usage: !MOVE (start), (end+1), (new start)

NOT ALLOWED: !MOVE 4096, 12288, 8192

OLD - un-news a NEWed program.

Usage: !OLD

BORDER - changes border color.

Usage: !BORDER (color)

BACK - same as border, but changes the background.

Usage: !BACK (color)

INC - increases a memory location.

Usage: !INC (location), (inc value)

!INC 53248,50 increases location 53248 by 50. BASIC equivalent is POKE 53248, PEEK(53248)+50. The advantage is that the value will never go over 255. So if a location has a value of 250 in it, and you INC it by 10, the value will wrap around to 5.

DEC - same function as INC, but decreases the location.

Usage: !DEC (location), (dec value)

DELETE - deletes a given number of lines.

Usage: !DELETE (starting line), (ending line)

SLEEP - pauses a program by a set number of seconds.

Usage: !SLEEP (number delay in seconds)

PAUSE - this will pause the program an extremely short time. This delays the program in JIFFIES (sixtieths of a second).

Usage: !PAUSE (number delay in 1/60 seconds)

POINTER - in immediate mode, this gives the memory location of the variable pointer. In program mode, it stores it at locations 40651 and 40652.

Usage: !POINTER (variable)

EOR - exclusive OR's the first number with the second. In immediate mode, prints the value, in program mode, stores it at 40653 and 40654.

Usage: !EOR (first number), (second number)

IMMEDIATE MODE: !EOR 7,9 - EOR's 7 with 9, prints value.

----- TEXT COMMANDS -----

CLS - clears part or all of the screen.

Usage: !CLS [starting row],[ending row]

LINECLR - clears a single line.

Usage: !LINECLR (row)

COLOR - changes cursor/print color.

Usage: !COLOR (color)

CURSOR - positions cursor at a certain location.

Usage: !CURSOR (row),(column)

PRINTAT - Prints at specific ROW and column on text screen.

Usage: !PRINTAT (row),(column)
!PRINTAT 24,13,"press a key..."

SPEED - controls PRINT speed. This only affects output going to the screen.

Usage: !SPEED (speed),(on/off)
The speed is from 1 to 255. A 1 turns it on, and a 0 turns it off.

CSET - copies the ROM character sets.

Usage: !CSET (upper/lowercase),(location)
A 0 signifies uppercase, a 1 lowercase.

CHAR - redefines a character. You must be using a RAM-based redefined character set or ROM set copied to RAM with the !CSET command.

Usage: !CHAR (screen code),(data 1),,,(data 8)

RECT - draws a box on the screen.

Usage: !RECT (rev),(color),(width),(height),(row),(column),(char)

The (rev) is for reverse or non-reverse characters. A 0 is for non-reverse, a 1 for reverse. To draw a reverse red box, with a height of 10 and width of 7, starting at row 4, column 9, using asterisks:

```
!RECT1,2,7,10,4,9,"*
```

NOTE: Don't try to make a line with this command. It is for boxes only.

FRAME - same as RECT, but draws a full border on the screen.

Usage: !FRAME (rev), (color), (char)

ANION - activates animation mode. You must be using a redefined character set.

Usage: !ANION (location of charset)

ANIOFF - turns off animation mode.

Usage: !ANIOFF

ANISPEED - sets the speed of animation.

Usage: !ANISPEED (speed value) Where the speed value is from 1-255.

ANIUP, ANIDOWN, ANILEFT, ANIRIGHT - lets you animate up to 4 characters, each in a different direction.

Usage: !ANIUP (screen code)

----- DISK COMMANDS -----

These are a set of disk commands to make certain functions a bit easier to use.

DEVICE - sets the device to the drive you'll be using. Drives 8-11 are supported, 8 is default. If a device is addressed that is not online, it will default back to 8.

Usage: !DEVICE (device)

@ - reads the error channel.

Usage: !@

DIR (also **CATALOG**) - lists the directory onto the screen without corrupting memory.

Usage: !DIR
!CATALOG

SEQ - reads a sequential file without corrupting any memory. NOTE: !SEQ also works with PRG and USR files.

Usage: !SEQ"filename"

INIT - initializes the drive.

Usage: !INIT

RESET - power up reset to the device in use.

Usage: !RESET

COLLECT - validates a disk.

Usage: !COLLECT

HEADER (also **FORMAT**) - either formats or does a short new to a disk.

Usage: !HEADER "disk name,id" or
!FORMAT "disk name,id" or
A\$="name,id":!FORMAT A\$

SCRATCH - scratches file(s) on a disk.

Usage: !SCRATCH (file 1,file 2,...)

RENAME - renames a file.

Usage: !RENAME"old filename"TO"new filename"

DLOAD - does a BASIC LOAD from the disk drive.

Usage: !DLOAD"filename"

DSAVE - does a BASIC save to the disk drive.

Usage: !DSAVE"filename"

RLOAD - same as DLOAD, but will also RUN program.

Usage: !RLOAD"filename"

BOOT - LOADs an ML program, then does a SYS to its LOAD address.

Usage: !BOOT"ml file"

ADD - LOADs a file from disk, and appends it to a file in memory. The file on disk must start with a higher line number than the last one of the program in memory.

Usage: !ADD"sub file"

LINESAVE - saves part of a BASIC program.

Usage: !LINESAVE"filename"(start line),(end line)

NOTE: When a file is LINESAVED, the line pointers will not be correct when it is LOADED back in. To correct this problem, after it is LOADED, type SYS 42291 to relink all the line pointers. Or you can simply hit RETURN over a line. Adding a line forces BASIC to correct links.

BLOAD - LOADs a binary file.

Usage: !BLOAD"file",[new LOAD address]

BSAVE - saves a binary file.

Usage: !BSAVE"file", (start), (end+1), [new LOAD address]

COMPLOAD - use the command for LOADING the compressed GG and JJ graphic files.

Usage: COMPLOAD"ggfilename"

----- **FUNCTION KEYS** -----

STAR BASIC allows macros for the function keys. It allows up to 12 macros, using the COMMODORE KEY to get keys 9 thru 12. When listing or redefining the keys, the BACK ARROW is used for RETURN, and the UP ARROW is used for SHIFT/RETURN.

DISABLE - disables the function keys.

Usage: !DISABLE

ENABLE - re-enables the function keys.

Usage: !ENABLE

LISTKEYS - lists the 12 keys and their macros.

Usage: !LISTKEYS

KEY - lets you define a macro for a key.

Usage: !KEY (key number), (macro string_)

----- **GRAPHICS** -----

These commands will let you use the hi-res and multicolor modes of the computer. When in these modes, the bitmap used is from 24576-32575 (\$6000-\$7F3F), and the video matrix is at 23552-24551 (\$5C00-\$5FE7). Bank 1 is in use, and the sprite pointers switch to \$5FF8-\$5FFF.

HIRES - puts the computer into hi-res mode.

Usage: !HIRES

MULTI - puts the computer into multi-color mode.

Usage: !MULTI NOTE: Whenever these two commands are used, the value at 53272 (\$D018) is saved, in case you're using a redefined character set.

TEXT - puts the computer back into text mode.

Usage: !TEXT

DDIS - displays a DOODLE picture LOAded at 23552 (\$5C00).

Usage: !DDIS

KDIS - displays a KOALA picture LOAded at 24576 (\$6000).

Usage: !KDIS

----- **SPRITE COMMANDS** -----

These commands will make sprites easier to use. No more POKEing or worrying about the dreaded MSB in case your sprite goes too far to the right.

SPRDEF - defines a sprite.

Usage:- !SPRDEF (sprite #), (mem block), (color), (xsize),
(ysize), (priority), (multicolor)

The sprite number is between 0 and 7. The memory block is the location of the sprite in memory divided by 64. Example: If you have a sprite in memory at 12288 (\$3000), the memory block would be 192, since 12288/64 is 192. The color is the individual color for the sprite (not any of the multi-colors). The X and Y size are in case you want to double its height or width. A 0 for normal, a 1 for double. The priority is for the sprite to appear either in front of or in back of graphics. A 0 puts the sprite in front, a 1 puts them in back. For the last number, a 0 denotes a hi-res (1 color) sprite, and a 1 denotes a multi-color sprite.

SPRMC - sets the 2 multi-color colors

Usage: !SPRMC (color 1), (color 2)

SPRPOS - sets the sprite's position

Usage: !SPRPOS (sprite #), (x loc), (y loc)

The sprite # is between 0 and 7, the X location (which would be 53248 for sprite #0), will be a value of 0-511. Anything over 255 will set the MSB for that particular sprite. Example: If the X value is 266, the X register would contain a 10, and the MSB would be set. The Y location is a value from 0 to 255.

SPRON - turns on individual sprites.

Usage: !SPRON (sprite #), (sprite #), , , !SPRON 0,1,2,3 - turns on the first 4 sprites.

SPROFF - turns off individual sprites.

Usage: !SPROFF - (sprite #), (sprite #),,,

SPRMOV - moves an individual sprite.

Usage: !SPRMOV (sprite #), (x step), (x loop), (x dir), (y step),
(y loop), (y dir), (delay in jiffies)

The step and loop refer to how many times (loop) the sprite will move (step) in what direction (dir).

0 = right or down

1 = left or up

An X step of 5 with an X loop of 10 would move the sprite 5 pixels 10 times, with a delay between each movement.

VCLR - this clears all the VIC registers pertaining to sprites, filling them with 0.

Usage: !VCLR

----- MUSIC COMMANDS -----

Except for the first command, all of these require the SID.OBJ file to be in memory. A way will be shown to tell whether or not it's in memory.

SCLR - clears all the SID registers.

Usage: !SCLR

SIDON - turns on the sid interrupt, but first it checks to see if the SID.OBJ file is in memory. This can be done with the following line:

```
!SIDON:IF PEEK(40908)=0 THEN PRINT"no sid.obj"
```

Usage: !SIDON

SIDOFF - turns off sid interrupt.

Usage: !SIDOFF

SIDLOC - lets the sid routine know where the '.MUS' sid file is in memory.

Usage: !SIDLOC (memory location)

SIDSTART - starts the sid playing.

Usage: !SIDSTART (repeat/non-repeat value)

A value of 0 will play the song once, any other value will cause the song to play over and over)

SIDWAIT - pauses a sid.

Usage: !SIDWAIT

SIDCONT - starts the song up where it stopped.

Usage: !SIDCONT

NOTE: When a song is done playing, you must use the SIDLOC command to set it up, then use the SIDSTART command if you want to play it again.

SIDOFF - turns off the sid interrupts.

Usage: !SIDOFF

Take a look at the demo and see all the neat stuff you can now do with STAR BASIC.