

VIC Programming

On this page you find graphics programming information in general, such as how to set up various graphics modes, how to draw on screen, or how to display and move a sprite. If you are looking for information on how to code certain demo effects, you should look in the [Demo Effects](#) section instead.

The general rule of thumb is that *if something is of more general use than just for demo effects, or if it is too simple or restricted to constitute a full demo effect in itself (such as opening the borders or achieving a specific custom screen mode through VIC trickery), you will find it here rather than in the demo coding section.*

There is also some information related to VIC programming in the [Game Coding](#) section, and there is some information on 3d math in the [maths](#) section.

Basics

- [Visible area](#) - How much of the overscan area actual monitors actually show
- [Pixel aspect ratio](#) - To sum it up, it's 0.9365:1 (X:Y) on PAL
- [Introduction to the VIC](#) - by Pasi 'Albert' Ojala (from C= Hacking 4).
- [Memory configuration](#) - See the memory configuration section to see how to select VIC bank and where to place sprites etc.
- [Interrupts and timing](#) - See the section on interrupts for information on VIC/raster interrupts.
- [Built in screen modes](#) - written and compiled by Oswald/Resource
- [Introduction to Sprites](#) - written and compiled by Oswald/Resource

Using built in screen modes

- [Convert ASCII/PETSCII to screencodes](#) by Mace/Secure
- [Applications of modified character sets](#)
- [Display a multicolor Koala Picture](#) by Richard Bayliss
- [Generic koala shower \(VIC settings calculator\)](#) by JTR/DMAgic


Sprites

See the general section on how to use sprites. It contains basic usage as well as sprite multiplexers and other things, including (some) demo effects with sprites.

- [Sprites](#) - General section on sprites

Drawing on the screen

- [Various techniques to calculate addresses fast, common screen formats for pixel graphics](#)
- [Rather small and rather fast screen offset component splitter](#) - by JackAsser/Instinct

- [Bresenham's line algorithm](#) - basic implementation - by Raf/Vulture Design
- [Bresenham's line algorithm 2](#) - by Hein
- [Lines](#) (bresenham, collecting pixels into one byte, runlength slice) - by Bitbreaker  (slope and sing the SID / CIA for slopes still missing)
- [Filling the vectors](#) - by Bitbreaker
- [Dots and Plots](#) - by TWW/CTR
- [Bresenham Circle Routine](#) - Implementation by Scout
- [Draw RAM to screen or color memory](#) - by malcontent
- [4x4 mode and circle routine](#) - by malcontent

VICII Hardware tricks

This section is more about the workings of the VIC chip than about certain demo effects which are *based* on these VIC tricks.

Border

- Opening upper and lower borders
 - [Short code](#) - by HCL/Booze Design
 - [Opening the borders](#) - by Richard Bayliss
 - [The long story](#) - by Pasi 'Albert' Ojala (C= Hacking #6)
 - [Opening up the borders - a further explanation](#) - by TheHighlander
 - [Opening borders and other things](#) - by Karmic/HVSC
- [Sprites in bottom sideborder](#) - by Groepaz/Hitmen
- [Removing the sideborders](#) - by HCL/Booze Design
- [Displaying graphics using \\$3fff](#)

Screen

Vertical Movement

- [FLD](#) - Flexible Line Distance - by HCL/Booze Design
- [Linecrunch](#) - Vertical Screen Positioning
- [Double Screen Vertical Scroller](#) - Using simple copy method with double buffer
- [Smooth Linecrunch](#) - by Karmic/HVSC/HF

Horizontal Movement

- [HSP](#) - Horizontal Screen Positioning (aka Wanker aka DMA-Delay/VSP - Variable Screen Positioning).

Useful Combinations

- [AGSP](#) - Any Given Screen Position (VSP and LineCrunch combined)
- [Double Screen Horizontal Scroller](#) - Using VSP to display a double bitmap picture (80×25 chars)

Vertical Tweaks

- [Introduction to vertical tweaks](#)
- [FLD](#) - Flexible Line Distance - by HCL/Booze Design
- [FPD](#) - Flexible Pixel Distance
- [FPS](#) - Flexible Pixel Size (like FPD but stretching pixels)
- [FPP](#) - Flexible Pixel Position
 - [FPP](#) - by repeating the last line
 - [FPP](#) - by repeating the first line
- [Repeating char-lines](#) - by HCL/Booze Design

NOTE: The names of above vertical movements are legacy and not always explicative. 'Line' in FLD refers to a char-line of 8 pixels high. 'Pixel' in FPD etc refers to a single line of graphics.

Sprites

This section about sprites..

Vertical Tweaks

- [Stretching sprites](#) - using \$d017.
- [Sprite FPD](#)
- [Sprite FPP](#)
- [Sprite-crunching](#)

Sprite Tweaks in general

* [9th Sprite](#) code and explanation

Custom Graphics Modes

This section is for custom graphics modes only, as contrasted to the built in graphics modes with official hardware support, such as hires, multicolor, etc.

Non-tweaked modes

- [SH](#) - Super Hires
- [Drazlace](#)
 - [Display Drazlace picture](#) - by HCL

Tweaked modes

- [FLI](#) - Flexible Line Interlace

- [FLI](#) - More color to the screen - by Pasi Ojala (C=Hacking #4).
- [FLI displayer](#) - by Graham
- [FLI displayer](#) - by Crossbow (only 40 bytes long)
- [Hires FLI](#) - High Resolution FLI
- [IFLI](#) - Interlaced FLI
- ([AFLI](#) - Advanced FLI)
- [UFLI](#) - Underlayer FLI
- [SHFLI](#) - Super Hires FLI
- [SHIFLI](#) - Super Hires Interlace FLI
- [XFLI](#) - EndOfNameSpace :) FLI

Others

- [Junk Modes](#) - low resolution modes with charsets
- [4x4 FLI chunky mode](#)

Editors for the Custom Graphics Modes

See the general section on:

- [GFX Editors and Converters](#)

Color info

- [VIC-II Color Cheatsheet](#) - by Groepaz
- [Commodore VIC-II Color Analysis](#) - by Pepto
 - Color fade (color ramps)
- by Veto

Graphics file formats

- [C64 Graphics File Format Specs](#) - by Groepaz

There is also a nice collection of file format info at AAY64:

- <http://unusedino.de/ec64/technical/aay/c64/frmmain.htm> - Graphics file formats

Various VIC related

- [VIC-Article](#) - The mighty... By Christian Bauer. (external link)
- [VIC register reference at AAY64](#) - Very handy online reference. By Ninja/The Dreams
- [Detect VIC Type](#) - by Crossbow
- [Detect PAL/NTSC](#) - Various routines
- [NTSC/PAL Differences](#) - article from Domination #13 written by Decomp/Style

From:
<https://codebase64.org/> - **Codebase 64 wiki**

Permanent link:
<https://codebase64.org/doku.php?id=base:vic>

Last update: **2017-03-30 11:27**

