

Contents

vdcmaniac	1
Contents	1
Inspiration	1
Screenshots	2
Known issues	7
Requirements	7
Building from source	8
License	9
Changelog	9

vdcmaniac

A modern Commodore 128 VDC (8563/8568, 80-column chip) demo, written from scratch in C with the [Oscar64](#) cross-compiler.

Contents

- [Inspiration](#)
- [Screenshots](#)
- [Known issues](#)
- [Requirements](#)
- [Building from source](#)
- [License](#)
- [Changelog](#)
- Further documentation: [ARCHITECTURE.md](#) (code layout), [vdc_reference_manual.md](#) (VDC registers/modes), [raster_lib_manual.md](#) (raster bar/IRQ library), [krill_manual.md](#) (fastloader integration)

Inspiration

This project is inspired by **VDC Mode Mania**, a 2012 demo by **Tokra** and **Mike of Akronym** **Analogiker**, originally released at the Connected 9 party (Sep 2, 2012), later extended twice (an 800x600 mode in 2015, a 960x540 mode in 2023). VDC Mode Mania was a slideshow showcasing the C128's rare, little-used 64KB-VDC-RAM graphics modes -- interlaced and non-interlaced colour and monochrome modes at resolutions far beyond the VIC-II's usual 320x200, from 480x252 colour up to 960x540 monochrome -- with converted picture assets and BASIC-based viewers. The original demo, including its BASIC source and image converters, is available from CSDb: <https://csdb.dk/release/?id=234174>

vdcmaniac is **not a port** of that BASIC codebase. It's a from-scratch reimplementaion in C, targeting the same family of rare VDC modes but built around this project's own tooling and techniques:

- **Oscar64**, a modern C cross-compiler for 6502 targets, instead of interpreted BASIC.
- **Krill's fastloader**, with on-the-fly TSCrunch decompression, for asset loading -- see [krill_manual.md](#).

- A **custom raster-IRQ trampoline**, installed directly on the CPU's hardware interrupt vector (not the KERNAL soft vector, which this project's own boot sequence makes unreliable), driving per-scanline VDC colour effects and, going forward, synced SID music playback -- see `raster_lib_manual.md`.
- A **from-scratch VDC mode/register library** (`include/vdc_core.c`) and its own picture converters (`tools/vdc_convert.py`) rather than the original's PC-side C converters -- see `vdc_reference_manual.md`.

The set of VDC modes being brought back is the same one VDC Mode Mania demonstrated (see the original's own README, linked above, for the full list and Tokra's own notes on real-hardware/monitor compatibility per mode); the techniques driving them are new.

Three further releases fed into specific techniques along the way:

- Peter Hulstede's "VDC-Intromaker: Perfektes Rasterzeilen-Timing", **64'er Sonderheft 95**, p.45 -- its example CIA-timer VDC raster-sync routine (`SYNC / WAITLINE / WAITJUMP / LINEND`, 6502 listing) is the basis this project's own `raster_synch() / raster_waitline()` (`include/vdc_raster.c`) are translated from into C, driving every raster-bar effect in the demo -- see that file's own credit comment for the full technical mapping.
- **"Risen from Oblivion VDC v2"** (Crest/Oxyron, 2006, <https://csdb.dk/release/?id=44983>) -- its own VDC-timing "system analysis" screen is where this project's `raster_calibrate()` (also `include/vdc_raster.c`) got the idea of measuring real cycles-per-VDC-rasterline on the running machine against the VDC status register's sync edge, rather than hardcoding a value. Independently reimplemented from that idea via a different mechanism (this project's own CIA1-timer-based measurement, not a code transplant) -- see the function's own credit comment for a note on what the original actually does internally.
- **"VDC SpectruMania"** (Tokra, Akronyme Analogiker, 2021, <https://csdb.dk/release/?id=206013>) and **"Colour Spectrum"** (Crest, concept by Tokra, code by JackAsser, loader by Krill, 2021, <https://csdb.dk/release/?id=205653>) -- both C128 VDC slideshows of reimagined ZX Spectrum graphics, and the reference points for the idea of showing Spectrum-sourced pictures on the C128's VDC ahead of this project's own from-scratch real-.scr-dump decode. See `include/defines.h`'s own credit block for the technical detail on what was verified against SpectruMania's conversion routine.

On pictures: every showcase section uses this project's own sourced artwork, converted via `tools/vdc_convert.py` -- public-domain paintings/ prints and CC-licensed photographs, each with a full attribution/licence note in `include/defines.h`'s own credit block. None of Tokra's original converted images are used.

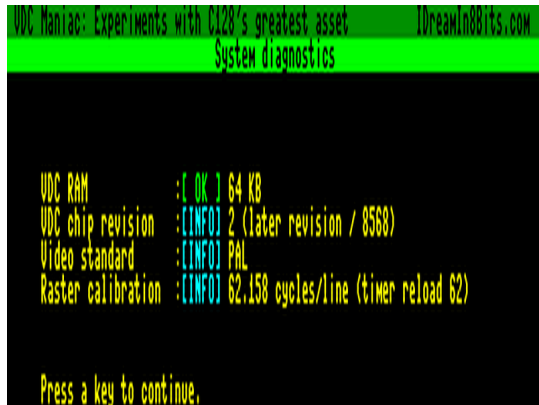
On music: the resident background tune is "Maniac" (Michael Sembello, from *Flashdance*, 1983), in a SID cover by Antti Hannula (Flex), 2010, Artline Designs -- credited in full in `include/defines.h`'s own credit block and in the demo's own end-credits scroller.

Screenshots

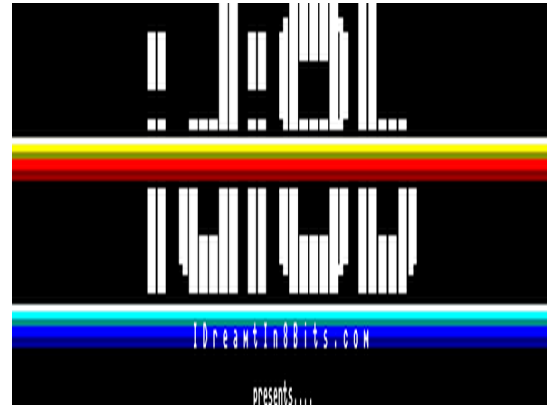
Captured in VICE (x128). VDC pixels are not square, so a raw capture misrepresents every mode's true on-screen proportions -- each screenshot below is cropped to that mode's own visible content and restored to a standard 4:3 aspect ratio. Full picture licence/attribution detail lives in

include/defines.h's own credit block; the captions below summarise it per picture.

Intro sequence



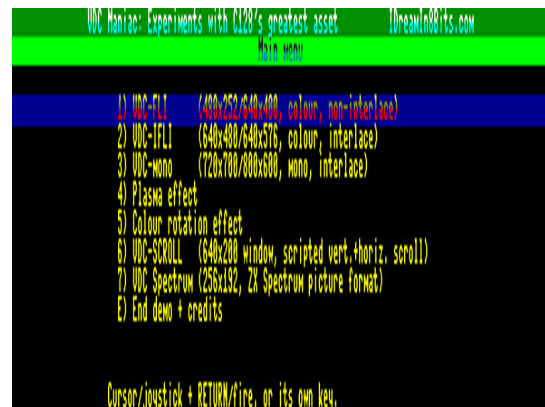
System diagnostics (PAL/NTSC, VDC revision, 64KB check)



idi8b logo section

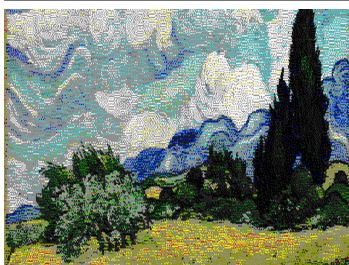


Title screen, raster bar underneath

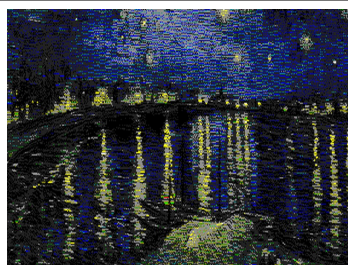


Main menu, raster-bar row highlight

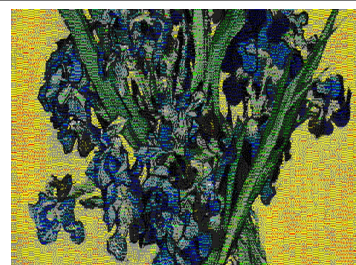
VDC-FLI (480x252, colour, non-interlace) + VDC-HFLI (640x400, colour, non-interlace)



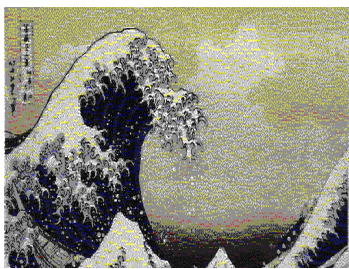
"Wheat Field with Cypresses"
(1889), Vincent van Gogh,
The Met, public domain



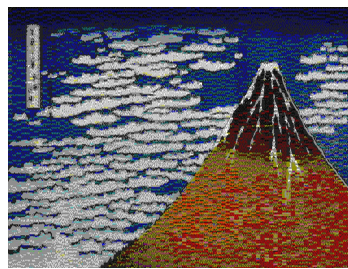
"Starry Night Over the
Rhone" (1888), Vincent van
Gogh, Musee d'Orsay, public
domain



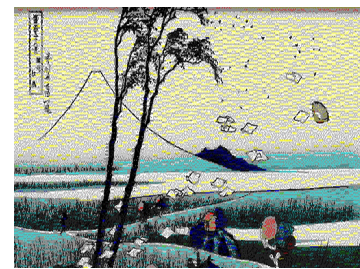
"Vase with Irises Against a
Yellow Background" (1890),
Vincent van Gogh, Van Gogh
Museum, public domain



"The Great Wave off Kanagawa" (c. 1831), Katsushika Hokusai, public domain

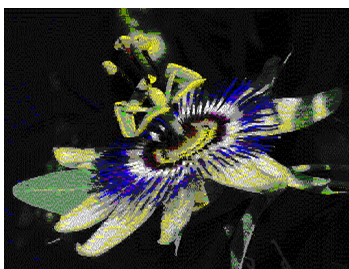


"Fine Wind, Clear Morning" ("Red Fuji", c. 1830-1832), Katsushika Hokusai, public domain



"Ejiri in Suruga Province" (c. 1830-1832), Katsushika Hokusai, public domain

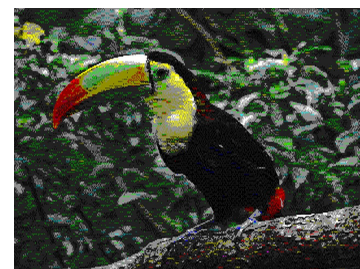
VDC-IFLI: VDC-IHFLI (640x480, colour, interlace) + VDC-ITFLI (640x576, colour, interlace)



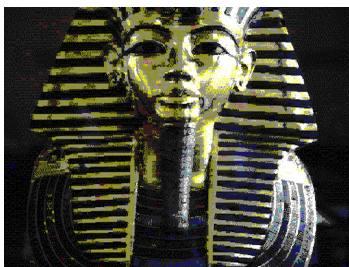
"Passiflora caerulea (makro close-up)" by Petar Milosevic, CC BY-SA 4.0



"Sonnenblume Helianthus 1" by Bohringer Friedrich, CC BY-SA 2.5



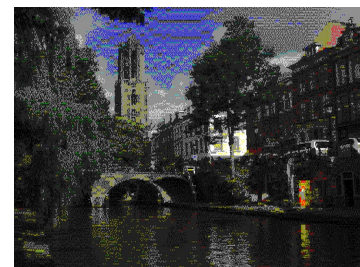
"Keel-billed Toucan, Caves Branch Jungle Lodge, Belize" by Judy Gallagher, CC BY 2.0



"Tutanchamun Maske" (funerary mask) by MykReeve, CC BY-SA 3.0 / GFDL 1.2+

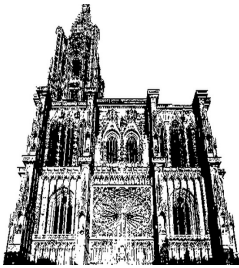


"Hyacinth macaw head", the Pantanal, Brazil, by Charles J. Sharp, CC BY-SA 4.0

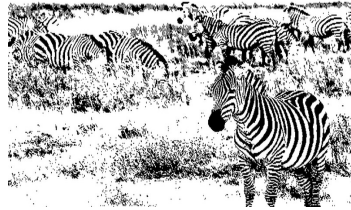


"De Hamburgerbrug met de Oudegracht en de Domtoren in de Stad Utrecht" by Jan dijckstra, CC BY-SA 4.0

VDC-mono: VDC-IMONO (720x700, interlace) + VDC-IM800 (800x600, interlace)



"Strasbourg Cathedral Exterior" by David Iliff (Diliff), CC BY-SA 3.0 / GFDL 1.2+



"Zebras Ngorongoro Crater" by Muhammad Mahdi Karim, GFDL 1.2



"Portrait de femme en tenue traditionnelle de Berbere Algerien" by Samia Dib Benkaci, CC BY-SA 4.0



"Portrait-of-a-woman" by Mark Sherman, CC BY 2.0

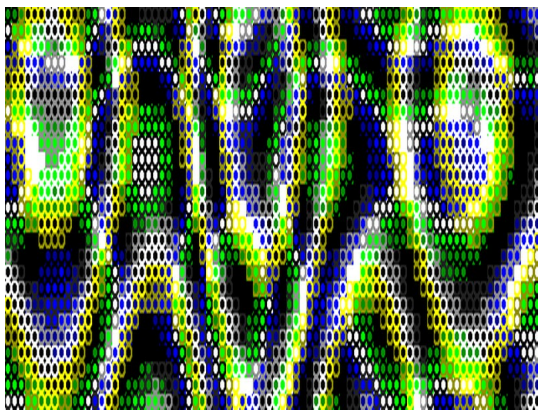


"The History of Apple Pie - Kelly Lee Owens (2013)" by Sylvain lasco, CC BY-SA 4.0

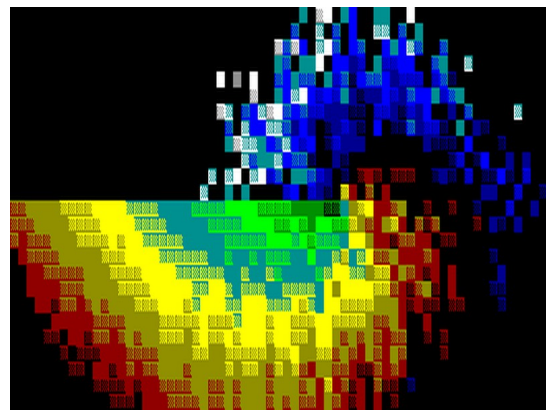


"Maupi", the author's own cat

Plasma effect / Colour rotation effect

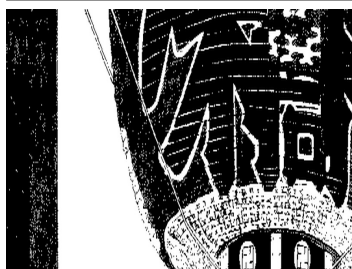


Sine-table plasma, own procedural content

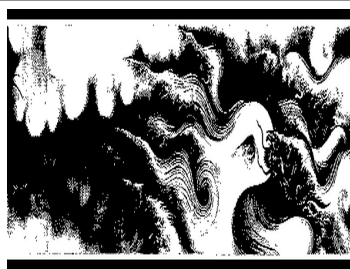


Colour-cycling bitmap, own procedural content

VDC-SCROLL: VDC-VSCROLL + VDC-PANORAMA + VDC-PANORAMA 2D



"Kinryuzan Temple, Asakusa", print #99 from One Hundred Famous Views of Edo (1856), Utagawa Hiroshige, public domain -- vertical scroll



"Nine Dragons" (1244), Chen Rong, Museum of Fine Arts, Boston, public domain -- horizontal pan

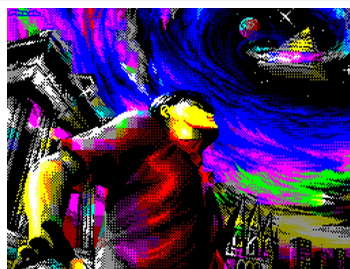


"The Last Stand of the Kusunoki at Shijonawate" (1857), Utagawa Kuniyoshi, British Museum, public domain -- combined 2D pan

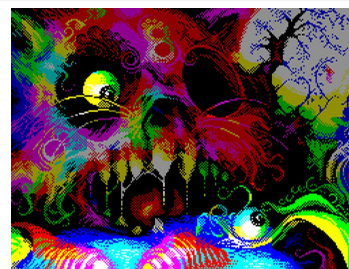
VDC Spectrum (256x192, doubled to 512x192)



"np" (2015), prof4d, 1st place at CC Winter - DiHalt Lite 2015



"Prisoner of Time" (2001), Pheel, 1st place at Chaos Constructions 2001



"Cursed Eighth" (2010), Piesiu, 1st place at the Chaos Constructions 2010 ZX Graphics compo

Two earlier Akronyme Analogiker/Crest C128 releases were reference points for the idea of showing ZX Spectrum-sourced graphics on the C128's VDC, ahead of this section's own from-scratch real-.scr-dump decode -- **"VDC SpectruMania"** (Tokra, Akronyme Analogiker, 2021, <https://csdb.dk/release/?id=206013>) and **"Colour Spectrum"** (Crest, concept by Tokra, code by JackAsser, loader by Krill, 2021, <https://csdb.dk/release/?id=205653>), 2nd place in the Mixed Demo Competition at Underground Conference 11. See include/defines.h's own credit block for the technical detail on what was verified against SpectruMania's conversion routine.

End demo + credits



Scrolling-text and colour-cycling-bars credits sequence.

Known issues

- **Monitor compatibility varies by VDC mode.** Not every one of the VDC's rare bitmap modes displays cleanly on every monitor -- this is a real-hardware VDC characteristic, not a bug in this demo (see the original VDC Mode Mania's own README, linked in "Inspiration" above, for Tokra's own notes on the same thing). A **Commodore 1901** monitor is confirmed to show every mode correctly, but may still need its own horizontal/vertical size and centering controls adjusted per mode -- as may any other monitor.
- **z64k:** the title screen renders mangled. Every other mode is unaffected. VICE shows every mode correctly.
- **RGBtoHDMI:** each VDC mode needs its own RGBtoHDMI mode definition. This project does not currently provide RGBtoHDMI mode definitions for any of its VDC modes.
- **VICE/z64k under WSL2/WSLg:** SID music playback can noticeably slow down in some sections -- a WSL2/WSLg performance characteristic, not reproduced on native Windows/Linux or real hardware. For accurate timing/audio, prefer running VICE/z64k natively rather than under WSL2.

Requirements

- **A VDC with the full 64 KB of RAM.** Every hires mode this demo showcases needs the extended 64KB VDC memory to exist at all -- a VDC with only the base 16 KB cannot run them. `system_diagnostic_screen()` checks this at startup and exits back to BASIC with a clear message if the VDC reports anything other than 64 KB. Most C128s have this by default; it's called out here because it's not universal across every 8563/8568 revision and board.
- **Krill's fastloader requires exclusive use of the IEC bus.** Only one device may be active on the bus: the drive holding the demo disk, at device ID 8. Power off (or otherwise disable) any other IEC device -- printers, second drives, etc. -- before running the demo.
- **Only a D81 disk image is supported.** D71 and especially D64 would need far more disk swapping to fit this project's own asset sizes, so neither is built or supported.

- **SD2IEC is not supported.** Krill's loader needs cycle-exact IEC bus timing, which SD2IEC's own IEC emulation does not provide. Use a real 1541/1571/1581 (or equivalent, e.g. an Ultimate II+) drive.

Building from source

Prerequisites

Tool	Purpose	Install
oscar64	C compiler targeting C128	Build from source or download release
c1541	Disk image creation (part of VICE)	<code>sudo apt install vice</code>
wput	FTP upload to Ultimate II+	<code>sudo apt install wput</code>
java (JRE 8+)	Run z64k (make z64k)	<code>sudo apt install default-jre</code> (optional)
pandoc + texlive-xetex	Regenerate README.pdf from README.md	<code>sudo apt install pandoc texlive-xetex</code> (optional)
python3 + Pillow	Regenerate converted picture assets	<code>pip install Pillow</code> (optional -- build warns and skips if missing)

The compiler is expected at `/home/xahmol/oscar64/bin/oscar64`. Edit the `CC` variable in the Makefile if yours is installed elsewhere.

Deployment setup (.env)

Create a `.env` file in the project root to configure deployment to your Ultimate II+. This file is gitignored and will never be committed:

```
ULTIP1 = 192.168.1.xx      # IP of your primary Ultimate II+
# ULTIP2 = 192.168.1.yy    # optional second machine
# ULTIP3 = 192.168.1.zz    # optional real-hardware test machine (make
↪  deploy3)
```

The Makefile constructs the FTP path for `deploy` / `deploy2` as `ftp://$(ULTIP1)/usb1/temp/`. Override `ULTUSB ?= usb1` in `.env` if your USB port is numbered differently. `deploy3` uses its own dedicated path (`ULTPATH3`, default `/USB1/idi8b/dev/`) instead -- override that in `.env` too if you want it elsewhere.

Make targets

Target	Description
make /make all	Build the program, boot sector and the Krill-fastloader D81 disk image (see krill_manual.md)
make krill	Same D81 build, invokable on its own
make clean	Remove all build artefacts
make vice	Launch VICE x128 with the D81 -- needs True Drive Emulation enabled (VICE Settings -> Drive, or x128 -drive8truedrive); Krill's loader runs drive-side install code (M-E / M-R) that silently hangs at the "loading assets" screen without it
make z64k	Launch z64k with the D81 (downloads Z64K.jar on first use; Java 8+ required) -- a second, independently-implemented C128/VDC emulator, useful when VICE doesn't reproduce a real-hardware issue
make deploy	Upload build to primary Ultimate II+ (requires ULTIP1 in .env)
make deploy2	Upload to second Ultimate II+ (requires ULTIP2 in .env)
make deploy3	Upload build to a real-hardware test machine (requires ULTIP3 in .env)
make docs	Regenerate README.pdf from README.md (requires pandoc)
make zip	Build build/vdcmaniac_<version>.zip -- the D81 plus README.pdf, for distribution

License

This project's own code is licensed under the [GNU General Public License v3.0](#). Krill's fastloader (krill/) and TSCrunch (krill/loader/tools/tscrunch/) are vendored third-party components under their own, separate, permissive attribution-required licenses -- both are credited in-demo (the end-credits scroller) and in this README's own Inspiration section, per their own license terms. Every picture/artwork/music asset has its own individual attribution and licence noted in include/defines.h's credit block -- see "On pictures" above.

Changelog

[v1.0.2](#)

Two threads: further raster_calibrate() cycle-precision refinements from Tokra, and a real-hardware-only VDC-IHFLI stall found and fixed via live diagnosis on the same Ultimate II+L test setup.

raster_calibrate() precision, from Tokra reviewing the routine directly against his own

cycle-exact PET-code VDC sync routine:

- **A third VBlank-edge check added before the routine's own sync point.** Entering the routine at an arbitrary point in the frame, the first `vdc_wait_vblank()` call could land within a few cycles of the vblank-to-display edge, making its result marginally late -- matches the "3 checks, since raster could be in vblank when checking or at end of non-vblank area" reasoning in Tokra's own sync routine.
- **CIA1 Timer B now configured/started before Timer A**, not after -- Timer B (in "count Timer A underflows" mode) is harmless to start before Timer A has begun, so starting Timer A last means none of Timer B's own setup cycles leak into the measured elapsed time.

Both are small precision tightenings ("nearly academic," Tokra's own words), not a systematic bias like v1.0.1's sync-point fix.

VDC-IHFLI ~15s real-hardware-only picture-transition stall, fixed:

- **Root cause:** `vdc_set_mode()` restored the destination mode's character set (`vdc_restore_charsets()`, a 512-byte, per-byte VDC-ready-pollled push) *before* applying that mode's own timing registers -- so the push ran while the *outgoing* mode's timing was still live on the chip. VDC-IHFLI has the narrowest real vertical-blanking window of any mode in this project's whole mode table, narrow enough that real 8563/8568 silicon (never VICE or z64k) could stall the push for real seconds. VDC-ITFLI, IHFLI's closest sibling, has a window nearly twice as wide -- almost certainly why only IHFLI ever showed it. Full technical write-up, including the live-diagnosis method used to isolate it, in `vdc_reference_manual.md`.
- **Fix:** the charset restore now runs after the mode's own timing registers are applied, so it always executes under safe (destination) timing regardless of what the outgoing mode was.
- **Also fixed along the way** (real, kept regardless of the above): CIA1 Control Register A's SPMODE bit (bit 6) -- which belongs to Krill's fastloader once it's installed, not this project -- was being silently clobbered by a couple of whole-byte CIA1 timer writes in `detect_ntsc()/raster_calibrate()`, which could have forced burst transfers on a drive that can't do them; and a redundant 64KB VDC memory wipe was removed from every picture-showcase section's own loop-top transition (`vdc_blank_pause()`, replacing `vdc_wipe_transition()` there), since `vdc_init()`'s own blank-then- clear-then-enable sequence already made it unnecessary.

v1.0.1

A real-hardware feedback pass, entirely prompted by **Tokra** (Akronyme Analogiker) live-testing v1.0.0 on his own C128 and RGBtoHDMI setup after release and reporting back in detail -- credited throughout the code's own comments at each specific fix, and here as a full, explicit thank-you for the time spent testing and the precision of the reports (register-level io d600 comparisons, not just "it looks wrong"):

- **VSYNC (register 7) off-by-one, VDC-IMONO/VDC-IM800/VDC-ITFLI:** Tokra reported RGBtoHDMI interlace fields swapped for the mono modes; a live VICE monitor register-dump comparison against his own working build (his own idea) found this project's VTOTAL - VAD-JUST derivation undershooting his real literal values by exactly 1 for three of the four interlaced modes he checked (VDC-IHFLI's own value already matched). Fixed to his own constants.

- **REFRESH (register 36) boot-baseline leak:** Tokra noticed register 36 deviating between the two builds during the same comparison pass; turned out to be this project's own well-known "boot-baseline register leak" bug class (see `vdc_reference_manual.md`) recurring on a register that hadn't been added to the capture/restore set yet.
- **Live VSYNC nudge (cursor up/down):** added directly on Tokra's own observation, from real-monitor testing, that "interlace is super fiddly -- some displays may actually need the +1 value," and that his own original demo already offers exactly this live adjustment.
- **raster_calibrate() sync-point bug:** Tokra reviewed a screenshot of the calibration loop directly and identified that the CIA timer started counting before any known VBlank-aligned reference point, letting the first sample land anywhere in a frame. Fixed by establishing that reference point before the timer starts; cross-validated afterward against an independent measurement (Risen from Oblivion VDC v2, on the same real hardware) landing on the identical value.
- **Title screen raster bar retuned** for the small timing shift the calibration fix introduced.

Also in this release, unrelated to Tokra's feedback: on-screen key hints for the VSYNC-nudge/colour-cycle keys, the running build version+timestamp now shown on the system diagnostics screen, and a genuine diagonal leg (not just alternating vertical/horizontal edges) added to VDC-PANORAMA 2D's own scripted tour.

v1.0.0

Initial public release: all eight VDC bitmap-mode showcases (VDC-FLI, VDC-HFLI, VDC-IHFLI, VDC-ITFLI, VDC-IMONO, VDC-IM800, VDC Spectrum, plus the Plasma and Colour rotation procedural effects), the scripted VDC-SCROLL family (vertical, horizontal, and combined 2D pan), a real-hardware-verified raster-IRQ engine, Krill-fastloader/TSCrunch asset loading, and complete documentation -- verified on real Commodore 128D/ 128DCR hardware with an Ultimate II+, not just in emulation.