



VIBE CODING MONITORS

How I Talked an AI Into Taming **SCORPIO's** Multi-GPU Display Nightmare

...and lived to write `DisplayRestore.ahk` about it

Presented by Steve Parker

October 14, 2026 | OPCUG Monthly Meeting

AI:

“How dumbed-down do you want this presentation?”

Steve:

“So I can understand it.”

AI:

“Oh, I’m sorry, I didn’t think you’d want it **THAT** dumb.”

I Asked AI To: Make HDR Work on my Monitor

A Hardware Triage that became a Vibe Coding exercise.

AI:

“We’ll get to this after we explain why.”

Steve:

“We will? Yes, WE will!”

1

Defining The Problem

We’ll get to the root cause — It’s a real **MUX-less** mess.

2

The HDR Solution had Three Unforeseen Side Effects

By solving one problem we had found 3 more!

3

Vibe Coding to the Rescue

Three Primary Automation Functions, One Script to bind them all

4

Lessons Learned & Why It Matters to You

Because a few of you may have this exact headache. *(See the Deleted Scenes)*



Meet SCORPIO: One Laptop, Two Personalities



ASUS TUF Gaming A15
computername SCORPIO

- AMD Ryzen 9 4900H with AMD Radeon Vega (iGPU)
- NVIDIA GeForce RTX 2060 (dGPU)
- **MUX-less** — no hardware switching between GPUs

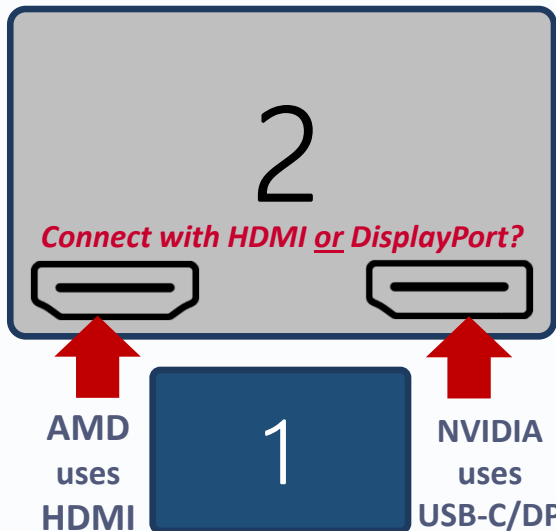
MUX-less = each GPU has a separate exit door.

Steve:

“MUX-less??? yeah, right, you just made that up.”

AI:

“No, I Didn’t. Look it up!”



“And here's the catch...”

NVIDIA only uses USB-C/DP

Samsung Odyssey Neo G8

- 32" curved 240Hz
- 4K **10bit HDR**
- EDID: SAM7256
- driven by the **NVIDIA dGPU** via USB-C/DP

Job: be gorgeous, demand everything.

Steve:

“Why is this important?”

AI:

“I’ll explain . . . LATER!”

Panda LM156LF-2F03

- Internal laptop panel
- EDID*: NCP004D
- driven by the AMD iGPU **NO 10bit HDR**

Job: exist quietly, then get upstaged.

*External Display Identification Data

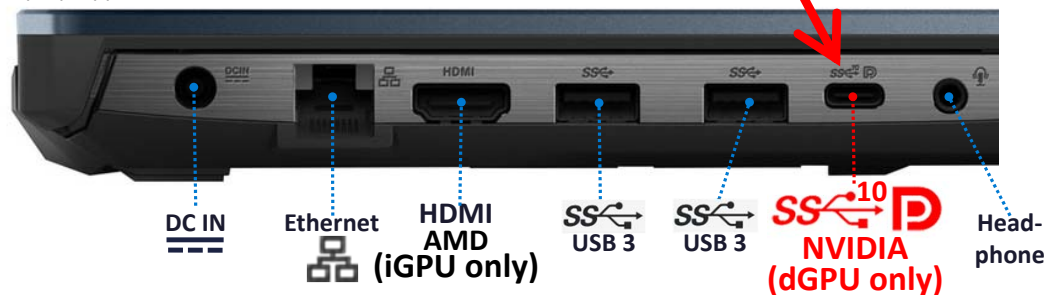
How SCORPIO's Displays Are Now Wired



ASUS TUF Gaming A15

AMD iGPU + NVIDIA RTX 2060 dGPU

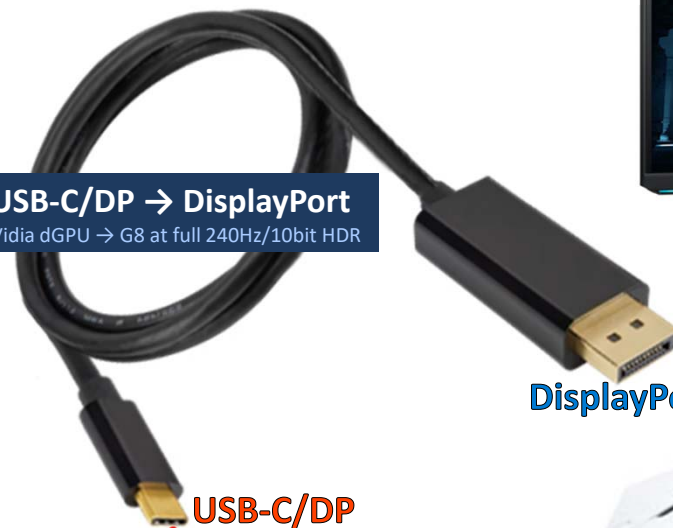
Laptop UpperLeft Side:



USB-C/DP → DisplayPort

nVidia dGPU → G8 at full 240Hz/10bit HDR

USB-C/DP



Samsung Odyssey Neo G8

Curved 32" 4K 240Hz/10bit HDR • EDID: SAM7256 • dGPU

DisplayPort



Monitor Under Side:

- | | | | | | |
|----------------|------------|------------------|-------|---------|-------|
| 1 HDMI IN1 | 2 HDMI IN2 | 3 DisplayPort IN | 4 USB | 5 PC IN | 6 USB |
| 7 Headphone IN | 8 DC 20V | | | | |

The Root Cause: The GPU Trap



Switching from HDMI to USB-C/DP solved the HDR problem

SCORPIO's laptop HDMI port is wired straight to the **AMD iGPU only** — the NVIDIA dGPU simply *cannot* output through it.

Steve:

“Well, this is another fine mess you’ve gotten me into!”

To get the G8's full 240Hz/HDR experience, the dGPU has to go out via USB-C/DP instead — a wiring quirk, not a setting.

By solving one problem we found 3 more!

Why this breaks everything

- Two GPUs now permanently split duty on one MUX-less laptop.
- Every sleep, wake, or lock forces Windows to renegotiate which GPU owns which screen — and Windows loses track of my window locations in the process.
- *This one wiring limitation is the seed of nearly every symptom in this talk.*

The HDR Solution had Three Unforeseen Side Effects

1



qBittorrent wanders off

A 90TB+ Torrent client that will not stay where it's put.

Windows:

"Where did you want that window?"

Steve:

"Right there."

Windows:

"Excellent. I'll put it somewhere else."

2



Windows don't come back after wakeup

Every sleep/lock scatters every open window at random.

3



Monitor ↔ TV handoff breaks settings

Windows doesn't properly keep display settings for two external screens.

*One AutoHotkey script,
two GPUs,
three displays,
and a laptop that genuinely cannot decide who's driving.*

DisplayRestore.ahk: One Script To Bind Them All

(DESCRIBE THE PROBLEM → GENERATE A DRAFT → TEST → CORRECT → REPEAT) ↺



DisplayRestore.ahk — Primary Automation Functions

Function 1

- Keeping qBittorrent in Place *Pinned coordinates + a Win+Q hotkey*

Function 2

- Auto-Restore After Wakeup *Session-lock snapshot + a debounced restore*

Function 3

- Monitor ↔ TV Handoff *Manual checklist + two dedicated hotkeys*

The Loop in Practice: Real Examples from This Project

Step	What Steve Said / Did	What Claude Produced	Result
Describe	“Windows scatters on every sleep/wake. qBittorrent keeps wandering off my screen.”	WinMove hotkey + WM_DISPLAYCHANGE handler to snapshot and restore windows	✓ Draft in minutes, not an evening with MSDN
Test + Fail	Restore fired too early — display still renegotiating. Windows scattered worse.	<i>(WM_DISPLAYCHANGE fires multiple times per change — no debounce yet)</i>	✗ Worse than before. Back to Claude.
Correct	“It fires too early — still scattering.” Reported the repeated-event problem.	Added 3-second debounced SetTimer. Waits for display stack to fully settle.	✓ Restores correctly. Running daily since June 2026.
Describe	“Don’t let Win+F2 restart Explorer before qBittorrent can shutdown.”	Safety check: if qBittorrent is running, cancel Win+F2 with a warning. Don’t touch Explorer.	✓ Claude missed it. Human caught it on real hardware.
Human Edge	Tested on real hardware. Caught the missing safety check. Rejected unsafe ideas.	<i>(Claude generated; Steve integrated, validated and decided what stayed)</i>	The judgment call is always human. Plus a long restore from the 90TB+ backup.

AI is fast at scaffolding. Humans are essential for testing, judgment, and knowing when it’s actually done.

Keeping qBittorrent in Place



`x = 0, y = 2160 • 1920×1080 • maximized` — plus a `Win+Q` hotkey to snap it back on demand



```
; From DisplayRestore.ahk — production
#q::
    WinGet, hwnd, ID, ahk_exe qbittorrent.exe
    WinRestore, ahk_id %hwnd%
    WinMove, ahk_id %hwnd%, 2160, 1920, 1080
    WinMaximize, ahk_id %hwnd%
```

WARNING: TECHNOBABBLE DETECTED

How it works

- The torrent client is nailed to one exact spot, always
- `Win+Q` is a manual override for whenever it still drifts
- A 90TB+ array doesn't care about “vibes” — predictability wins

Auto-Restore After Wakeup



On session lock: snapshot every window. On **WM_DISPLAYCHANGE**: wait, then restore.

```
; From DisplayRestore.ahk – producti
OnMessage(0x7E, "OnDisplayChar

WM_DISPLAYCHANGE(w
SetTimer
Repeat, MoveQBSTSilent()
repeat restores
```

WARNING: YET STILL MORE TECHNOBABBLE

How it works

- Windows fires this event several times per real change
- The debounce waits for things to settle before acting
- Then every saved window position gets put back, once

FUNCTION 3

Monitor ↔ TV Handoff *(THE TV WAS PRETENDING TO BE THE LAPTOP)*



A friend's 65" TV ("Beyond TV") reports the **exact same EDID** as the laptop's own panel — hardware can't tell them apart, so this handoff needs a human-followed checklist plus two hotkeys.



Windows:

"I see two identical displays."

Steve:

"They're not identical."

Windows:

"They are to me."



```
; From DisplayRestore.ahk — production code
#F2:: ApplyG8Settings()
ApplyG8Settings() {
    RunWait, regedit.exe /s G8_DisplaySettings.reg
    Sleep, 2000 ; let display stack settle
    RunWait, taskkill /f /im explorer.exe
```

The two hotkeys

Win + F1 — backs up the WindowMetrics key before switching

Win + F2 — restores the G8's registry values and restarts Explorer

The Night The Fonts Shrank



There is more to this but that's another Vibe.

VIBE OF THE APPXCHECKER

COMING SOON

An NVIDIA driver update shrank every icon and font system-wide — 150% scale rendering like **~10pt instead of 12pt.**

Cleared suspects:

- *WindowMetrics*
- *PerMonitorSettings*
- *AppliedDP*
- *Win+F2 restore*
- *GPU scaling*
- *a warm reboot*

Claude AI:

“Check each of these in order.”

Steve:

“I’ll just try shutting down first.”

Claude AI:

“Smarty Pants.”

The only cure: a full shutdown, completely powered off for 5 minutes.

A *MUX-less dGPU* can get so stuck mid-DPI-negotiation with the monitor, that only a real power-cycle can reset it.

A Little Extra Hardening



Teaching Windows to Leave a Note

ApplyG8OnShutdown.xml, a Task Scheduler task on Event ID 1074 (shutdown), stamps the correct G8 registry values before power-off so DPI is right on the next boot.



Demoting the Redundant Coworker

RefreshRateService.exe was quietly trying to manage the same display profiles DisplayRestore.ahk already handled — so it got set to Manual.

It Works!

The last edit of **DisplayRestore.ahk** was made on June 12/2026.

*It's run flawlessly, all by itself, several times a day, since then.
It still makes me smile to watch it in operation.*

What AI Got Right & Wrong



It's confidently wrong, a lot

The AI will hand you a debounce-free event handler with total conviction
— Test everything; be patiently persistent.



It's a fantastic first-draft machine

Hotkeys, snapshot logic, registry scaffolding
— minutes instead of an evening with MSDN.



The weirdest bugs aren't code bugs

Duplicate EDIDs and DPI negotiation deadlocks are hardware having a personality, not hallucinations.



Knowledge $\neq$ Wisdom

It has the knowledge but lacks the wisdom to use it properly. That's where ***you*** come in.



Thank You, OPCUG!

Questions?

I'll answer them exactly like an AI would — confidently, and only sort of right.

Steve Parker | October 14, 2026 | OPCUG

DisplayRestore.ahk — three functions, one script to bind them all

Bonus thought: Most people with big expensive TVs are plugging in HDMI, seeing the HDR badge light up — and not getting true HDR. They just don't know it yet. (See Deleted Scenes)



Claude AI:

"I found a font called **Great Vibe**,
Do you want me to change it."

Steve:

"Don't you dare!"

**Coming Up Next:
Deleted Scenes**

DELETED SCENES

DisplayRestore.ahk • The Outtakes

TAKE: ∞

ROLL: 1

OUTTAKE 1 — The Saturn V Request

Steve: "Could you create a super detailed ultraphotorealistic Saturn V?"

Claude: *"That's outside what I can do here — I can't generate images."*

OUTTAKE 2 — The Furniture Incident

Steve: "From NanoBanana comes the BigBanana"

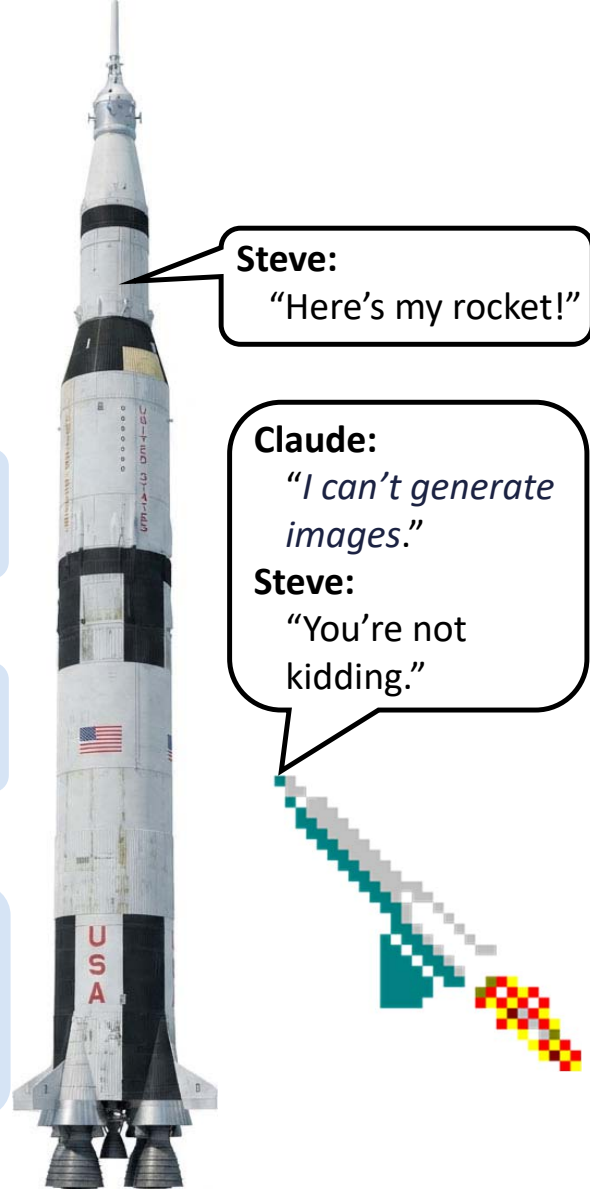
Claude: *"I love it — but I need a bit more context before I start moving furniture."*

OUTTAKE 3 — The Rocket's Departure

Claude: "Was the rocket causing a specific problem, or just didn't fit?"

Steve: "'Functions' was more appropriate than 'Stages' so the rocket had to go."

Claude: "Ha — sometimes the narrative wins over the visual. 'Function 1/2/3' is definitely cleaner and more accurate, even if the rocket was a great gag."



OUTTAKE 4 — WHY HDR MAY NOT WORK

Steve: “So, most people with mega-sized-expensive TVs are still not getting true HDR

Claude: “Almost certainly, yes — and most of them don't even know it.

The typical setup is laptop HDMI → TV, which means iGPU → 8-bit dithered → “HDR” mode enabled in Windows → TV shows the HDR badge on screen → user thinks they have HDR. And it looks *pretty good* because dithering is clever — it's only in smooth gradients like skies, sunsets, or dark scenes where the banding betrays it.

The TV manufacturers know this too. The HDR badge lights up based on the signal *type* being flagged as HDR, not whether the content is actually true 10-bit. So, the TV is being honest about what it received, but what it received was already compromised upstream at the GPU. (*cont'd on next slide*)

OUTTAKE 4 — WHY HDR MAY NOT WORK — (Cont'd)

Claude: It gets worse when you factor in that most people with gaming laptops are in the same MUX-less boat you were in — HDMI hardwired to the iGPU, no way to get the dGPU's output through that port. The ones who *do* get true HDR are typically:

- Using a desktop with a discrete GPU and a direct HDMI 2.1 or DisplayPort
 - Using a laptop with a MUX switch that can route the dGPU to the HDMI port
 - Using USB-C/DP like you are now
- but only if they figured out that's even possible

The vast majority just plug in HDMI, see the HDR badge, and assume it's working.

That's actually a quietly powerful point for your OPCUG talk — you started trying to fix HDR and discovered that most people's "HDR" probably isn't. That reframes the whole project from "one person's weird laptop problem" to something the audience might go home and check on their own setup.