The Mystery of the Purple Line on the Bally Astrocade Menu

If you have ever powered on a Bally Astrocade, you may have noticed a thin purple vertical line along the right edge of the game menu. The same line also appears when running the system in the MAME emulator. It is subtle, but once you notice it, it is hard to unsee.



A Small Quirk in the System Power-Up Routine

While examining the Astrocade BIOS source code, I noticed that during system startup the UPI routine `SETOUT` is called with a horizontal color boundary value of **41**, even though the hardware supports a range that extends to **42**.

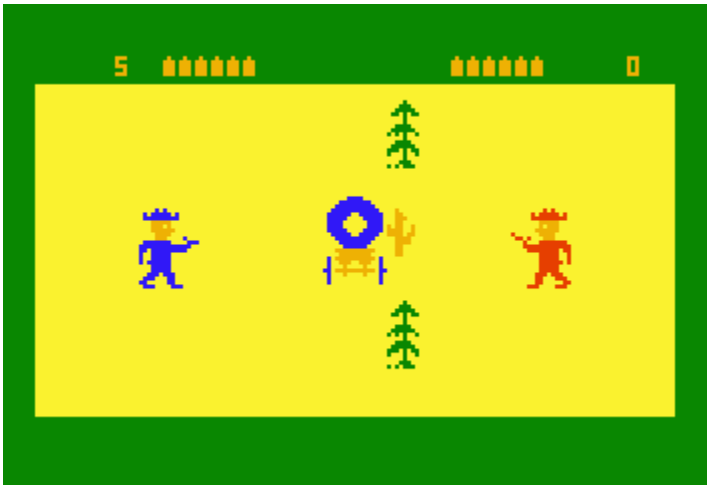
At first glance, this looks like a simple off-by-one issue. However, the fact that this behavior exists on real hardware and persists across multiple BIOS revisions suggests it was either intentional or at least consciously accepted by the original designers.

How the Horizontal Color Boundary Works

The horizontal color boundary controls how colors are applied across the screen in blocks of 4 pixels. The effective range is **0 to 42**.

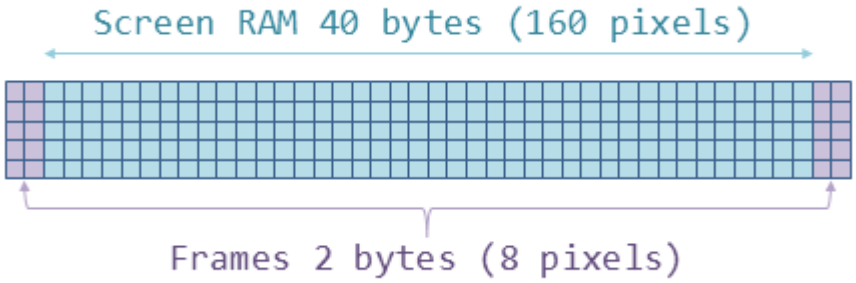
This mechanism exists to effectively **double the number of colors available on a single scanline**. Although the Astrocade is limited to **four colors at any given horizontal position**, the boundary allows one set of four colors to be used on the left side of the screen and a different set of four colors on the right side. By moving the boundary, software can divide the screen into two color regions, each with its own palette. This works very well for games like Gunfight where there are

5 colors on the screen (Green, Yellow, Beige, Blue and Red). Player Two's outfit is red because the right side of the screen has a different palette with red replacing blue.



The Astrocade screen RAM contains **40 bytes per row**. Because the system uses 2-bit color, each byte represents **4 pixels**, resulting in a visible horizontal resolution of **160 pixels**.

In addition to the usable screen area, the hardware adds a fixed frame on both the left and right sides of the display. Each side of the frame is **2 bytes wide**, or **8 pixels**. You cannot draw graphics into the frame area, but you can select which of the four system colors it uses.



An Asymmetrical Hardware Detail

One unusual quirk of the Astrocade hardware is that the **right-side frame** is included in the horizontal color boundary calculation, while the left-side frame is not. This makes the effective horizontal span:

```
40 bytes (screen RAM) + 2 bytes (right-side frame) = 42
```

Because of this, the boundary must be set to **42** to cleanly stop color assignment at the visible screen edge. When the boundary is set to **41**, the final 4-pixel column on the right edge draws its color from the right-side palette instead of the left.

Why the Line Is Purple

To save ROM space, the system power-up routine defines only a **4-byte color table** labelled `MENUCL`. However, the `COLSET` routine expects **8 bytes** total: 4 bytes for the left side and 4 bytes for the right side.

Since only 4 bytes are provided, the color set routine continues reading whatever data follows the table in memory.

Immediately following in the BIOS is:

```
DB $FF

JP RVECT3
```

Which assembles to:

```
$FF $C3 $0D $20
```

Because the frame color defaults to zero column register, which is processed last, the right-side frame ends up using the value **$20** for one 4-pixel column. On the Astrocade, **$20 corresponds to purple**, which explains the vertical purple line seen on the menu screen.

Why It May Have Been Left That Way

Although it is tempting to label this as a bug, there are several reasonable explanations for why this behavior survived multiple BIOS revisions.

On many CRT televisions of the era, the far-right edge of the video signal fell into the **overscan region** and was not visible to the player. The purple line sits right at that edge and likely would not have appeared on many consumer TVs at all.

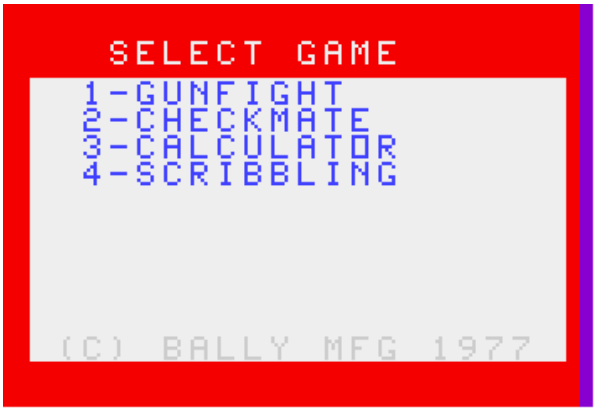
It is also possible the line had some internal value during development or quality control. On service monitors or lab displays with little or no overscan, a visible color sliver at the extreme right edge could act as a useful reference for horizontal alignment, frame behavior, or color boundary timing. I suspect this is the real reason.

And finally, it may simply have been regarded as harmless. If it did not affect gameplay, was usually hidden by overscan, and the menu behaved correctly, there may have been little incentive to change it once the system shipped.

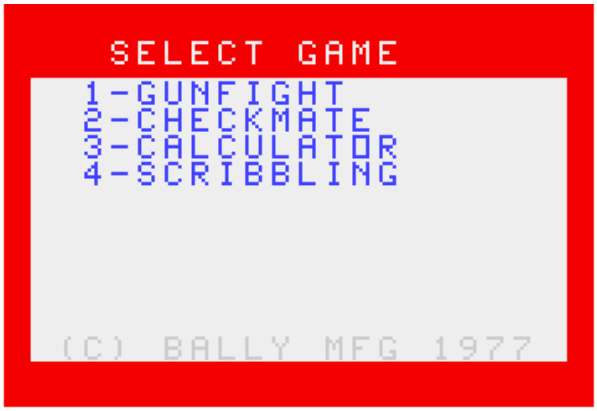
A Tiny Detail with Staying Power

This is ultimately an academic curiosity, not something that needs to be fixed. Still, it is a neat example of how small hardware details, subtle assumptions about display technology, and a few adjacent bytes of code can combine to produce a visible artifact that persists on real hardware and in modern emulation.

Below are images showing the original BIOS behavior and a patched version with the corrected boundary value.



original



patched

rdagger

1/5/2026